- For security; to allow packet filtering at firewalls.
- For forwarding at the initial router (when hosts don't do MPLS)

Callon et al.   Expires March 2000   [Page 24]
□
IETF Draft   A Framework for MPLS   September 1999

Consider a campus network which is serving a small company. Suppose that this company makes use of the Internet, for example as a method of communicating with customers. A customer on the other side of the world has an IP packet to be forwarded to a particular system within the company. It is not reasonable to expect that the customer will have a label to use to forward the packet to that specific system. Rather, the label used for the "first hop" forwarding might be sufficient to get the packet considerably closer to the destination. However, the granularity of the labels cannot be to every host worldwide. Similarly, routing used within one routing domain cannot know about every host worldwide. This implies that in may cases the labels assigned to a particular packet will be sufficient to get the packet close to the destination, but that at some points along the path of the packet the IP header will need to be examined to determine a finer granularity for forwarding that packet. This is particularly likely to occur at domain boundaries.

A similar point occurs at the last router prior to the destination host. In general, the number of hosts attached to a network is likely to be great enough that it is not feasible to assign a separate label to every host. Rather, as least for routing within the destination routing domain (or the destination area if there is a hierarchical routing protocol in use) a label may be assigned which is sufficient to get the packet to the last hop router. However, the last hop router will need to examine the IP header (and particularly the destination IP address) in order to forward the packet to the correct destination host.

Packet filtering at firewalls is an important part of the operation of the Internet. While the current state of Internet security may be considerably less advanced than may be desired, nonetheless some security (as is provided by firewalls) is much better than no security. We expect that packet filtering will continue to be important for the foreseeable future. Packet filtering requires examination of the contents of the packet, including the IP header. This implies that at firewalls the packet cannot be forwarded simply by considering the label associated with the packet. Note that this is also likely to occur at domain boundaries.

Finally, it is very likely that many hosts will not implement MPLS. Rather, the host will simply forward an IP packet to its first hop router. This first hop router will need to examine the IP header prior to forwarding the packet (with or without a label).

3.2 Scaling Issues

MPLS scalability is provided by two of the principles of routing.

Callon et al.   Expires March 2000   [Page 25]

The first is that forwarding follows an inverted tree rooted at a destination. The second is that the number of destinations is reduced by routing aggregation.

The very nature of IP forwarding is a merged multipoint-to-point tree. Thus, since MPLS mirrors the IP network layer, an MPLS node that is capable of merging is capable of creating O(n) switched paths which provide network reachability to all "n" destinations. The meaning of "n" depends on the granularity of the switched paths. One obvious choice of "n" is the number of CIDR prefixes existing in the forwarding table (this scales the same as today's routing). However, the value of "n" may be reduced considerably by choosing switched paths of further aggregation. For example, by creating switched paths to each possible egress node, "n" may represent the number of egress nodes in a network. This choice creates "n" switched paths, such that each path is shared by all CIDR prefixes that are routed through the same egress node. This selection greatly improves scalability, since it minimizes "n", but at the same time maintains the same switching performance of CIDR aggregation. (See section 2.2.2 for a description of all of the levels of granularity provided by MPLS).

The MPLS technology must scale at least as well as existing technology. For example, if the MPLS technology were to support ONLY host-to-host switched path connectivity, then the number of switched-paths would be much higher than the number of routing table entries.

There are several ways in which merging can be done in order to allow O(n) switches paths to connect n nodes. The merging approach used has an impact on the amount of state information, buffering, delay characteristics, and the means of control required to coordinate the trees. These issues are discussed in more detail in section 4.2.

There are some cases in which O(n-squared) switched paths may be used (for example, by setting up a full mesh of point to point streams). As label space and the amount of state information that can be supported may be limited, it will not be possible to support O(n-squared) switched paths in very large networks. However, in some cases the use of n-squared paths may even be a advantage (for example, to allow load- splitting of individual streams).

MPLS must be designed to scale for O(n). O(n) scaling allows MPLS domains to scale to a very large scale. In addition, if best effort service can be supported with O(n) scaling, this conserves resources (such as label space and state information) which can be used for supporting advanced services such as QoS. However, since some switches may not support merging, and some small networks may

not require the scaling benefits of O(n), provisions must also be provided for a non-merging, O(n-squared) solution.

Note: A precise and complete description of scaling would consider
that there are multiple dimensions of scaling, and multiple
resources whose usage may be considered. Possible dimensions of
scaling include: (i) the total number of streams which exist in an
MPLS domain (with associated labels assigned to them); (ii) the
total number of "label swapping pairs" which may be stored in the
nodes of the network (ie, entries of the form "for incoming label
'x', use outgoing label 'y'"); (iii) the number of labels which
need to be assigned for use over a particular link; (iv) The
amount of state information which needs to be maintained by any
one node. We do not intend to perform a complete analysis of all
possible scaling issues, and understand that our use of the terms
"O(n)" and "O(n-squared)" is approximate only.

3.3 Types of Streams

Switched paths in the MPLS network can be of different types:

    - point-to-point
    - multipoint-to-point
    - point-to-multipoint
    - multipoint-to-multipoint

Two of the factors that determine which type of switched path is
used are (i) The capability of the switches employed in a network;
(ii) The purpose of the creation of a switched path; that is, the
types of flows to be carried in the switched path. These two
factors also determine the scalability of a network in terms of
the number of switched paths in use for transporting data through
a network.

The point-to-point switched path can be used to connect all
ingress nodes to all the egress nodes to carry unicast traffic. In
this case, since an ingress node has point-to-point connections to
all the egress nodes, the number of connections in use for
transporting traffic is of O(n-squared), where n is the number of
edge MPLS devices. For small networks the full mesh connection
approach may suffice and not pose any scalability problems.
However, in large enterprise backbone or ISP networks, this will
not scale well.

Point-to-point switched paths may be used on a host-to-host or
application to application basis (e.g., a switched path per RSVP
flow). The dedicated point-to-point switched path transports the
unicast data from the ingress to the egress node of the MPLS
network. This approach may be used for providing QoS services or
for best-effort traffic.

Callon et al.   Expires March 2000   [Page 27]
☐
IETF Draft   A Framework for MPLS   September 1999

A multipoint-to-point switched path connects all ingress nodes to
an single egress node. At a given intermediate node in the
multipoint-to-point switched path, L2 data units from several
upstream links are "merged" into a single label on a downstream
link. Since each egress node is reachable via a single multipoint-
to-point switched path, the number of switched paths required to
transport best-effort traffic through a MPLS network is O(n),
where n is the number of egress nodes.

The point-to-multipoint switched path is used for distributing

multicast traffic. This switched path tree mirrors the multicast distribution tree as determined by the multicast routing protocols. Typically a switch capable of point-to-multipoint connection replicates an L2 data unit from the incoming (parent) interface to all the outgoing (child) interfaces. Standard ATM switches support such functionality in the form of point-to-multipoint VCs or VPs.

A multipoint-to-multipoint switched path may be used to combine multicast traffic from multiple sources into a single multicast distribution tree. The advantage of this is that the multipoint-to-multipoint switched path is shared by multiple sources. Conceptually, a form of multipoint-to-multipoint can be thought of as follows: Suppose that you have a point to multipoint VC from each node to all other nodes. Suppose that any point where two or more VCs happen to merge, you merge them into a single VC or VP. This would require either coordination of VCI spaces (so that each source has a unique VCI within a VP) or VC merge capabilities. The applicability of similar concepts to MPLS is FFS.

3.4 Data Driven versus Control Traffic Driven Label Assignment

A fundamental concept in MPLS is the association of labels and network layer routing. Each LSR must assign labels, and distribute them to its forwarding peers, for traffic which it intends to forward by label switching. In the various contributions that have been made so far to the MPLS WG we identify three broad strategies for label assignment; (i) those driven by topology based control traffic [RFC2105][ARIS][IPNAV]; (ii) Those driven by request based control traffic [CR-LDP][RSVP-LSP]; and (iii) those driven by data traffic [RFC2098][RFC1953].

We also note that in actual practice combinations of these methods may be employed. One example is that topology based methods for best effort traffic plus request based methods for support of RSVP.

3.4.1 Topology Driven Label Assignment

Callon et al.   Expires March 2000   [Page 28]
□
IETF Draft   A Framework for MPLS   September 1999

In this scheme labels are assigned in response to normal processing of routing protocol control traffic. Examples of such control protocols are OSPF and BGP. As an LSR processes OSPF or BGP updates it can, as it makes or changes entries in its forwarding tables, assign labels to those entries.

Among the properties of this scheme are:

- The computational load of assignment and distribution and the bandwidth consumed by label distribution are bounded by the size of the network.

- Labels are in the general case preassigned. If a route exists then a label has been assigned to it (and distributed). Traffic may be label swapped immediately it arrives, there is no label setup latency at forwarding time.

- Requires LSRs to be able to process control traffic load only.

- Labels assigned in response to the operation of routing
  protocols can have a granularity equivalent to that of the
  routes advertised by the protocol. Labels can, by this
  means, cover (highly) aggregated routes.

3.4.2 Request Driven Label Assignment

In this scheme labels are assigned in response to normal
processing of request based control traffic. Examples of such
control protocols are RSVP. As an LSR processes RSVP messages it
can, as it makes or changes entries in its forwarding tables,
assign labels to those entries.

Among the properties of this scheme are:

- The computational load of assignment and distribution and
  the bandwidth consumed by label distribution are bounded by
  the amount of control traffic in the system.

- Labels are in the general case preassigned. If a route
  exists then a label has been assigned to it (and
  distributed). Traffic may be label swapped immediately it
  arrives, there is no label setup latency at forwarding time.

- Requires LSRs to be able to process control traffic load
  only.

- Depending upon the number of flows supported, this approach
  may require a larger number of labels to be assigned
  compared with topology driven assignment.

Callon et al.   Expires March 2000   [Page 29]
□
IETF Draft   A Framework for MPLS   September 1999

- This approach requires applications to make use of request
  paradigm in order to get a label assigned to their flow.

3.4.3 Traffic Driven Label Assignment

In this scheme the arrival of data at an LSR "triggers" label
assignment and distribution. Traffic driven approach has the
following characteristics.

- Label assignment and distribution costs are a function of
  traffic patterns. In an LSR with limited label space that is
  using a traffic driven approach to amortize its labels over
  a larger number of flows the overhead due to label
  assignment and distribution grows as a function of the
  number of flows and as a function of their "persistence".
  Short lived but recurring flows may impose a heavy control
  burden.

- There is a latency associated with the appearance of a
  "flow" and the assignment of a label to it. The documented
  approaches to this problem suggest L3 forwarding during this
  setup phase, this has the potential for packet reordering
  (note that packet reordering may occur with any scheme when
  the network topology changes, but traffic driven label
  assignment introduces another cause for reordering).

- Flow driven label assignment requires high performance
  packet classification capabilities.

- Traffic driven label assignment may be useful to reduce
  label consumption (assuming that flows are not close to full
  mesh).

- If you want flows to hosts, due to limits on label space,
  then traffic based label consumption is probably necessary
  due to the large number of hosts which may occur in a
  network.

- If you want to assign specific network resources to
  specific labels, to be used for support of application
  flows, then again the fine grain associated with labels may
  require data based label assignment.

3.5 The Need for Dealing with Looping

   Routing protocols which are used in conjunction with MPLS will in
   many cases be based on distributed computation. As such, during
   routing transients, these protocols may compute forwarding paths
   which contain loops. For this reason MPLS will be designed with

Callon et al.   Expires March 2000   [Page 30]
□
IETF Draft   A Framework for MPLS   September 1999

   mechanisms to either prevent the formation of loops and /or
   contain the amount of resources that can be consumed due to the
   presence of loops.

   Note that there are a number of different alternative mechanisms
   which have been proposed (see section 4.3). Some of these prevent
   the formation of layer 2 forwarding loops, others allow loops to
   form but minimize their impact in one way or another (e.g., by
   discarding packets which loop, or by detecting and closing the
   loop after a period of time). Generally speaking, there are
   tradeoffs to be made between the amount of looping which might
   occur, and other considerations such as the time to convergence
   after a change in the paths computed by the routing algorithm.

   We are not proposing any changes to normal layer 3 operation, and
   specifically are not trying to eliminate the possibility of
   looping at layer 3. Transient loops will continue to be possible
   in IP networks. Note that IP has a means to limit the damage done
   by looping packets, based on decrementing the IP TTL field as the
   packet is forwarded, and discarding packets whose TTL has expired.
   Dynamic routing protocols used with IP are also designed to
   minimize the amount of time during which loops exist.

   The question that MPLS has to deal with is what to do at L2. In
   some cases L2 may make use of the same method that is used as L3.
   However, other options are available at L2, and in some cases
   (specifically when operating over ATM or Frame Relay hardware) the
   method of decrementing a TTL field (or any similar field) is not
   available.

   There are basically two problems caused by packet looping: The
   most obvious problem is that packets are not delivered to the
   correct destination. The other result of looping is congestion.
   Even with TTL decrementing and packet discard, there may still be
   a significant amount of time that packets travel through a loop.

This can adversely affect other packets which are not looping:
Congestion due to the looping packets can cause non-looping
packets to be delayed and/or discarded.

Looping is particularly serious in (at least) three cases: One is
when forwarding over ATM. Since ATM does not have a TTL field to
decrement, there is no way to discard ATM cells which are looping
over ATM subnetworks. Standard ATM PNNI routing and signaling
solves this problem by making use of call setup procedures which
ensure that ATM VCs will never be setup in a loop [PNNI]. However,
when MPLS is used over ATM subnets, the native ATM routing and
signaling procedures may not be used for the full L2 path. This
leads to the possibility that MPLS over ATM might in principle
allow packets to loop indefinitely, or until L3 routing
stabilizes. Methods are needed to prevent this problem.

Callon et al.   Expires March 2000   [Page 31]
□
IETF Draft   A Framework for MPLS   September 1999

Another case in which looping can be particularly unpleasant is
for multicast traffic. With multicast, it is possible that the
packet may be delivered successfully to some destinations even
though copies intended for other destinations are looping. This
leads to the possibility that huge numbers of identical packets
could be delivered to some destinations. Also, since multicast
implies that packets are duplicated at some points in their path,
the congestion resulting from looping packets may be particularly
severe.

Another unpleasant complication of looping occurs if the
congestion caused by the loop interferes with the routing
protocol. It is possible for the congestion caused by looping to
cause routing protocol control packets to be discarded, with the
result that the routing protocol becomes unstable. For example
this could lengthen the duration of the loop.

In normal operation of IP networks the impact of congestion is
limited by the fact that TCP backs off (ie, transmits
substantially less traffic) in response to lost packets. Where the
congestion is caused by looping, the combination of TTL and the
resulting discard of looping packets, plus the reduction in
offered traffic, can limit the resulting impact on the network.
TCP backoff however does not solve the problem if the looping
packets are not discarded (for example, if the loop is over an ATM
subnetwork where TTL is not used).

The severity of the problem caused by looping may depend upon
implementation details. Suppose, for instance, that ATM switching
hardware is being used to provide MPLS switching functions. If the
ATM hardware has per-VC queuing, and if it is capable of providing
fair access to the buffer pool for incoming cells based on the
incoming VC (so that no one incoming VC is allowed to grab a
disproportionate number of buffers), this looping might not have a
significant effect on other traffic. If the ATM hardware cannot
provide fair buffer access of this sort, however, then even
transient loops may cause severe degradation of the node's total
performance.

Given that MPLS is a relatively new approach, it is possible that
looping may have consequences which are not fully understood (such
as looping of LDP control information in cases where stream merge

is not used).

Even if fair buffer access can be provided, it is still worthwhile
to have some means of detecting loops that last "longer than
possible". In addition, even where TTL and/or per-VC fair queuing
provides a means for surviving loops, it still may be desirable
where practical to avoid setting up LSPs which loop.

Callon et al.   Expires March 2000   [Page 32]
▢
IETF Draft   A Framework for MPLS   September 1999

Methods for dealing with loops are discussed in section 4.3.

3.6 Operations and Management

Operations and management of networks is critically important.
This implies that MPLS must support operations, administration,
and maintenance facilities at least as extensive as those
supported in current IP networks.

In most ways this is a relatively simple requirement to meet.
Given that all MPLS nodes run normal IP routing protocols, it is
straightforward to expect them to participate in normal IP network
management protocols.

There is one issue which has been identified and which needs to be
addressed by the MPLS effort: There is an issue with regard to
operation of Traceroute over MPLS networks. Note that other O&M
issues may be identified in the future.

Traceroute is a very commonly used network management tool.
Traceroute is based on use of the TTL field: A station trying to
determine the route from itself to a specified address transmits
multiple IP packets, with the TTL field set to 1 in the first
packet, 2 in the second packet, etc.. This causes each router
along the path to send back an ICMP error report for TTL exceeded.
This in turn allows the station to determine the set of routers
along the route. For example, this can be used to determine where
a problem exists (if no router responds past some point, the last
router which responds can become the starting point for a search
to determine the cause of the problem).

When MPLS is operating over ATM or Frame Relay networks there is
no TTL field to decrement (and ATM and Frame Relay forwarding
hardware does not decrement TTL). This implies that it is not
straightforward to have Traceroute operate in this environment.

There is the question of whether we *want* all routers along a
path to be visible via traceroute. For example, an ISP probably
doesn't want to expose the interior of their network to a
customer. However, the issue of whether a network's policy will
allow the interior of the network to be visible should be
independent of whether is it possible for some users to see the
interior of the network. Thus while there clearly should be the
possibility of using policy mechanisms to block traceroute from
being used to see the interior of the network, this does not imply
that it is okay to develop protocol mechanisms which prevent
traceroute from working.

There is also the question of whether the interior of a MPLS

network is analogous to a normal IP network, or whether it is
closer to the interior of a layer 2 network (for example, an ATM
subnet). Clearly IP traceroute cannot be used to expose the
interior of an ATM subnet. When a packet is crossing an ATM
subnetwork (for example, between an ingress and an egress router
which are attached to the ATM subnet) traceroute can be used to
determine the router to router path, but not the path through the
ATM switches which comprise the ATM subnet. Note here that MPLS
forms a sort of "in between" special case:
Routing is based on normal IP routing protocols, the equivalent of
call setup (label binding/exchange) is based on MPLS-specific
protocols, but forwarding is based on normal L2 ATM forwarding.
MPLS therefore supersedes the normal ATM-based methods that would
be used to eliminate loops and/or trace paths through the ATM
subnet.

It is generally agreed that Traceroute is a relatively "ugly"
tool, and that a better tool for tracing the route of a packet
would be preferable. However, no better tool has yet been designed
or even proposed. Also, however ugly Traceroute may be, it is
nonetheless very useful, widely deployed, and widely used. In
general, it is highly preferable to define, implement, and deploy
a new tool, and to determine through experience that the new tool
is sufficient, before breaking a tool which is as widely used as
traceroute.

Methods that may be used to either allow traceroute to be used in
an MPLS network, or to replace traceroute, are discussed in
section 4.11.

4. Technical Approaches

4.1 Label Distribution

A fundamental requirement in MPLS is that an LSR forwarding label
switched traffic to another LSR apply a label to that traffic
which is meaningful to the other (receiving the traffic) LSR.
LSR's could learn about each other's labels in a variety of ways.
We call the general topic "label distribution".

4.1.1 Explicit Label Distribution

Explicit label distribution anticipates the specification by MPLS
of a standard protocol for label distribution. Two of the possible
approaches (TDP, ARIS [ARIS-PROT]) are oriented toward topology
driven label distribution. One other approach [FANP], in contrast,
makes use of traffic driven label distribution. We expect that the
label distribution protocol [LDP] which emerges from the MPLS WG
is likely to inherit elements from one or more of the possible

approaches.

Consider LSR A forwarding traffic to LSR B. We call A the upstream
(wrt to dataflow) LSR and B the downstream LSR. A must apply a
label to the traffic that B "understands". Label distribution must
ensure that the "meaning" of the label will be communicated
between A and B. An important question is whether A or B (or some
other entity) allocates the label.

In this discussion we are talking about the allocation and
distribution of labels between two peer LSRs that are on a single
segment of what may be a longer path. A related but in fact
entirely separate issue is the question of where control of the
whole path resides. In essence there are two models; by analogy to
upstream and downstream for a single segment we can talk about
ingress and egress for an LSP (or to and from a label switching
"domain"). In one model a path is setup from ingress to egress and
in the other from egress to ingress.

4.1.1.1 Downstream Label Allocation

"Downstream Label Allocation" refers to a method where the label
allocation is done by the downstream LSR, ie the LSR that uses the
label as an index into its switching tables.

This is, arguably, the most natural label allocation/distribution
mode for unicast traffic. As an LSR builds its routing tables (we
consider here control driven allocation of tags) it is free,
within some limits we will discuss, to allocate labels in any
manner that may be convenient to the particular implementation.
Since the labels that it allocates will be those upon which it
subsequently makes forwarding decisions we assume implementations
will perform the allocation in an optimal manner. Having allocated
labels the default behavior is to distribute the labels (and
bindings) to all peers.

In some cases (particularly with ATM) there may be a limited
number of labels which may be used across an interface, and/or a
limited number of label assignments which may be supported by a
single device. Operation in this case may make use of "on demand"
label assignment. With this approach, an LSR may for example
request a label for a route from a particular peer only when its
routing calculations indicate that peer to be the new next hop for
the route.

4.1.1.2 Upstream Label Allocation

"Upstream Label Allocation" refers to a method where the label
allocation is done by the upstream LSR. In this case the LSR
choosing the label (the upstream LSR) and the LSR which needs to

Callon et al.   Expires March 2000   [Page 35]
□
IETF Draft   A Framework for MPLS   September 1999

interpret packets using the label (the downstream LSR) are not the
same node. We note here that in the upstream LSR the label at
issue is not used as an index into the switching tables but rather
is found as the result of a lookup on those tables.

The motivation for upstream label allocation comes from the
recognition that it might be possible to optimize multicast
machinery in an LSR if it were possible to use the same label on

all output ports for which a particular multicast packet/cell were
destined. Upstream assignment makes this possible.

4.1.1.3 Other Label Allocation Methods

Another option would be to make use of label values which are
unique within the MPLS domain (implying that a domain-wide
allocation would be needed). In this case, any stream to a
particular MPLS egress node could make use of the label of that
node (implying that label values do not need to be swapped at
intermediate nodes).

With this method of label allocation, there is a choice to be made
regarding the scope over which a label is unique. One approach is
to configure each node in an MPLS domain with a label which is
unique in that domain. Another approach is to use a truly global
identifier (for example the IEEE 48 bit identifier), where each
MPLS-capable node would be stamped at birth with a truly globally
unique identifier. The point of this global approach is to
simplify configuration in each MPLS domain by eliminating the need
to configure label IDs.

4.1.2 Piggybacking on Other Control Messages

While we have discussed use of an explicit MPLS LDP we note that
there are several existing protocols that can be easily modified
to distribute both routing/control and label information. This
could be done with any of OSPF, BGP, RSVP and/or PIM. A particular
architectural elegance of these schemes is that label distribution
uses the same mechanisms as are used in distribution of the
underlying routing or control information.

When explicit label distribution is used, the routing computation
and label distribution are decoupled. This implies a possibility
that at some point you may either have a route to a specific
destination without an associated label, and/or a label for a
specific destination which makes use of a path which you are no
longer using. Piggybacking label distribution on the operation of
the routing protocol is one way to eliminate this decoupling.

Piggybacking label distribution on the routing protocol introduces
an issue regarding how to negotiate acceptable label values and

Callon et al.   Expires March 2000   [Page 36]
□
IETF Draft   A Framework for MPLS   September 1999

what to do if an invalid label is received. This is discussed in
section 4.1.3.

4.1.3 Acceptable Label Values

There are some constraints on which label values may be used in
either allocation mode. Clearly the label values must lie within
the allowable range described in the encapsulation standards that
the MPLS WG will produce. The label value used must also, however,
lie within a range that the peer LSR is capable of supporting. We
imagine that certain machines, for example ATM switches operating
as LSRs may, due to operational or implementation restrictions,
support a label space more limited than that bounded by the valid
range found in the encapsulation standard. This implies that an
advertisement or negotiation mechanism for useable label range may
be a part of the MPLS LDP. When operating over ATM using ATM

forwarding hardware, due to the need for compatibility with the existing use of the ATM VPI/VCI space, it is quite likely that an explicit mechanism will be needed for label range negotiation.

In addition we note that LDP may be one of a number of mechanism used to distribute labels between any given pair of LSRs. Clearly where such multiple mechanisms exist care must be taken to coordinate the allocation of label values. A single label value must have a unique meaning to the LSR that distributes it.

There is an issue regarding how to allow negotiation of acceptable label values if label distribution is piggybacked with the routing protocol. In this case it may be necessary either to require equipment to accept any possible label value, or to configure devices to know which range of label values may be selected. It is not clear in this case what to do if an invalid label value is received as there may be no means of sending a NAK.

A similar issue occurs with multicast traffic over broadcast media, where there may be multiple nodes which receive the same transmission (using a single label value). Here again it may be "non-trivial" how to allow n-party negotiation of acceptable label values.

4.1.4 LDP Reliability

The need for reliable label distribution depends upon the relative performance of L2 and L3 forwarding, as well as the relationship between label distribution and the routing protocol operation.

If label distribution is tied to the operation of the routing protocol, then a reasonable protocol design would ensure that labels are distributed successfully as long as the associated route and/or reachability advertisement is distributed

Callon et al.   Expires March 2000   [Page 37]
□
IETF Draft   A Framework for MPLS   September 1999

successfully. This implies that the reliability of label distribution will be the same as the reliability of route distribution.

If there is a very large difference between L2 and L3 forwarding performance, then the cost of failing to deliver a label is significant. In this case it is important to ensure that labels are distributed reliably. Given that LDP needs to operate in a wide variety of environments with a wide variety of equipment, this implies that it is important for an LDP developed by the MPLS WG to ensure reliable delivery of label information.

Reliable delivery of LDP packets may potentially be accomplished either by using an existing reliable transport protocol such as TCP, or by specifying reliability mechanisms as part of LDP (for example, the reliability mechanisms which are defined in IDRP could potentially be "borrowed" for use with LDP).

TCP supports flow control (in addition to supporting reliable delivery of data). Flow control is a desirable feature which will be useful for MPLS (as well as other applications making use of a reliable transport) and therefore needs to be built into whatever reliability mechanism is used for MPLS.

4.1.5 Label Purge Mechanisms

Another issue to be considered is the "lifetime" of label data
once it arrives at an LSR, and the method of purging label data.
There are several methods that could be used either separately, or
(more likely) in combination.

One approach is for label information to be timed out. With this
approach a lifetime is distributed along with the label value. The
label value may be refreshed prior to timing out. If the label is
not refreshed prior to timing out it is discarded. In this case
each lifetime and timer may apply to a single label, or to a group
of labels (e.g., all labels selected by the same node).

Similarly, two peer nodes may make use of an MPLS peer keep-alive
mechanism. This implies exchange of MPLS control packets between
neighbors on a periodic basis. This in general is likely to use a
smaller timeout value than label value timers (analogous to the
fact that the OSPF HELLO interval is much shorter than the OSPF
LSA lifetime). If the peer session between two MPLS nodes fails
(due to expiration of the associated timer prior to reception of
the refresh) then associated label information is discarded.

If label information is piggybacked on the routing protocol then
the timeout mechanisms would also be taken from the associated
routing protocol (note that routing protocols in general have

Callon et al.   Expires March 2000   [Page 38]
▯
IETF Draft   A Framework for MPLS   September 1999

mechanisms to invalidate stale routing information).

An alternative method for invalidating labels is to make use of an
explicit label removal message.

4.2 Stream Merging

In order to scale O(n) (rather than O(n-squared)), MPLS makes use
of the concept of stream merge. This makes use of multipoint to
point streams in order to allow multiple streams to be merged into
one stream.

4.2.1 Types of Stream Merge:

There are several types of stream merge that can be used,
depending upon the underlying media.

When MPLS is used over frame based media merging is
straightforward. All that is required for stream merge to take
place is for a node to allow multiple upstream labels to be
forwarded the same way and mapped into a single downstream label.
This is referred to as frame merge.

Operation over ATM media is less straightforward. In ATM, the data
packets are encapsulated into an ATM Adaptation Layer, say AAL5,
and the AAL5 PDU is segmented into ATM cells with a VPI/VCI value
and the cells are transmitted in sequence. It is contingent on ATM
switches to keep the cells of a PDU (or with the same VPI/VCI
value) contiguous and in sequence. This is because the device that
reassembles the cells to re-form the transmitted PDU expects the
cells to be contiguous and in sequence, as there isn't sufficient
information in the ATM cell header (unlike IP fragmentation) to

reassemble the PDU with any cell order. Hence, if cells from
several upstream link are transmitted onto the same downstream
VPI/VCI, then cells from one PDU can get interleaved with cells
from another PDU on the outgoing VPI/VCI, and result in corruption
of the original PDUs by mis-sequencing the cells of each PDU.

The most straightforward (but erroneous) method of merging in an
ATM environment would be to take the cells from two incoming VCs
and merge them into a single outgoing VCI. If this was done
without any buffering of cells then cells from two or more packets
could end up being interleaved into a single AAL5 frame. Therefore
the problem when operating over ATM is how to avoid interleaving
of cells from multiple sources.

There are two ways to solve this interleaving problem, which are
referred to as VC merge and VP merge.

VC merge allows multiple VCs to be merged into a single outgoing

Callon et al.   Expires March 2000   [Page 39]
□
IETF Draft   A Framework for MPLS   September 1999

VC. In order for this to work the node performing the merge needs
to keep the cells from one AAL5 frame (e.g., corresponding to an
IP packet) separate from the cells of other AAL5 frames. This may
be done by performing the SAR function in order to reassemble each
IP packet before forwarding that packet. In this case VC merge is
essentially equivalent to frame merge. An alternative is to buffer
the cells of one AAL5 frame together, without actually
reassembling them. When the end of frame indicator is reached that
frame can be forwarded. Note however that both forms of VC merge
generally require that the entire AAL5 frame be received before
any cells corresponding to that frame be forwarded. VC merge
therefore requires capabilities which are generally not available
in most existing ATM forwarding hardware.

The alternative for use over ATM media is VP merge. Here multiple
VPs can be merged into a single VP. Separate VCIs within the
merged VP are used to distinguish frames (e.g., IP packets) from
different sources. In some cases, one VP may be used for the tree
from each ingress node to a single egress node.

4.2.2 Interoperation of Merge Options:

If some nodes support stream merge, and some nodes do not, then it
is necessary to ensure that the two types of nodes can
interoperate within a single network. This affects the number of
labels that a node needs to send to a neighbor. An upstream LSR
which supports Stream Merge needs to be sent only one label per
forwarding equivalence class (FEC). An upstream neighbor which
does not support Stream Merge needs to be sent multiple labels per
FEC. However, there is no way of knowing a priori how many labels
it needs. This will depend on how many LSRs are upstream of it
with respect to the FEC in question.

If a particular upstream neighbor does not support stream merge,
it is not known a priori how many labels it will need. The
upstream neighbor may need to explicitly ask for labels for each
FEC. The upstream neighbor may make multiple such requests (for
one or more labels per request). When a downstream neighbor
receives such a request from upstream, and the downstream neighbor
does not itself support stream merge, then it must in turn ask its

downstream neighbor for more labels for the FEC in question.

It is possible that there may be some nodes which support merge, but have a limited number of upstream streams which may be merged into a single downstream stream. Suppose for example that due to some hardware limitation a node is capable of merging four upstream LSPs into a single downstream LSP. Suppose however, that this particular node has six upstream LSPs arriving at it for a particular Stream. In this case, this node may merge these into two downstream LSPs (corresponding to two labels that need to be

Callon et al.   Expires March 2000   [Page 40]
□
IETF Draft   A Framework for MPLS   September 1999

obtained from the downstream neighbor). In this case, the node will need to obtain the required two labels.

The interoperation of the various forms of merging over ATM is most easily described by first describing the interoperation of VC merge with non-merge.

In the case where VC merge and non-merge nodes are interconnected the forwarding of cells is based in all cases on a VC (ie, the concatenation of the VPI and VCI). For each node, if an upstream neighbor is doing VC merge then that upstream neighbor requires only a single outgoing VPI/VCI for a particular FEC (this is analogous to the requirement for a single label in the case of operation over frame media). If the upstream neighbor is not doing merge, then it will require a single outgoing VPI/VCI per FEC for itself (assuming that it can be an ingress node), plus enough outgoing VPI/VCIs to map to incoming VPI/VCIs to pass to its upstream neighbors. The number required will be determined by allowing the upstream nodes to request additional VPI/VCIs from their downstream neighbors.

A similar method is possible to support nodes which perform VP merge. In this case the VP merge node, rather than requesting a single VPI/VCI or a number of VPI/VCIs from its downstream neighbor, instead may request a single VP (identified by a VPI). Furthermore, suppose that a non-VP-merge node is downstream from two different VP merge nodes. This node may need to request one VPI/VCI (for traffic originating from itself) plus two VPs (one for each upstream node).

An alternative method is possible, which requires no support of VP switching and VP labels on nodes which do not support VP merge. In this method, the VP merge node does not request VPs from the downstream node. It does request a number of VPI/VCIs, one per source node in the group of nodes which use VP merge.

In order to support all of VP merge, VC merge, and non-merge, it is therefore necessary to allow upstream nodes to request a combination of zero or more VC identifiers (consisting of a VPI/VCI), plus zero or more VPs (identified by VPIs). In addition, it may be helpful to allow upstream nodes to request zero of more VPs (identified by VPIs). VP merge nodes would therefore request one VP, or in the event where this is not supported, the VP merge node(s) would request several VCs . VC merge node would request only a single VPI/VCI (since they can merge all upstream traffic into a single VC). Non-VP-merge nodes would pass on any requests that they get from above, plus request a VPI/VCI for traffic that they originate (if they can be ingress nodes). However, non-merge

nodes which can only do VC forwarding (and not VP forwarding) will
need to know which VCIs are used within each VP in order to

Callon et al.   Expires March 2000   [Page 41]
□
IETF Draft   A Framework for MPLS   September 1999

install the correct VCs in its forwarding table. This limitation
is likely to apply to most on-ATM LSRs; most ATM NICs can
terminate VP connections as numbers individual VC connections. A
detailed description of how this could work can be found in
[ATMVP]. Alternatively, the non-VP-merge nodes could issue only VC
identifiers, as described above.

4.2.3 Coordination of the VCI space with VP Merge:

VP merge requires that the VCIs be coordinated to ensure
uniqueness. There are a number of ways in which this may be
accomplished:

1. Each node may be pre-configured with a unique VCI value
   (or values).

2. Some one node (most likely they root of the multipoint to
   point tree) may coordinate the VCI values used within the
   VP. A protocol mechanism will be needed to allow this to
   occur. How hard this is to do depends somewhat upon
   whether the root is otherwise involved in coordinating the
   multipoint to point tree. For example, allowing one node
   (such as the root) to coordinate the tree may be useful
   for purposes of coordinating load sharing (see section
   4.10). Thus whether or not the issue of coordinating the
   VCI space is significant or trivial may depend upon other
   design choices which at first glance may have appeared to
   be independent protocol design choices.

3. Other unique information such as portions of a class B or
   class C address may be used to provide a unique VCI value.

4. Another alternative is to implement a simple hardware
   extension in the ATM switches to keep the VCI values
   unique by dynamically altering them to avoid collision.

VP merge makes less efficient use of the VPI/VCI space (relative
to VC merge). When VP merge is used, the LSPs may not be able to
transit public ATM networks that don't support SVP.

4.2.4 Buffering Issues Related To Stream Merge:

There is an issue regarding the amount of buffering required for
frame merge, VC merge, and VP merge. Frame merge and VC merge
requires that intermediate points buffer incoming packets until
the entire packet arrives. This is essentially the same as is
required in traditional IP routers.

VP merge allows cells to be transmitted by intermediate nodes as
soon as they arrive, reducing the buffering and latency at

Callon et al.   Expires March 2000   [Page 42]
□
IETF Draft   A Framework for MPLS   September 1999

intermediate nodes. However, the use of VP merge implies that cells from multiple packets will arrive at the egress node interleaved on separate VCIs. This in turn implies that the egress node may have somewhat increased buffering requirements. To a large extent egress nodes for some destinations will be intermediate nodes for other destinations, implying that increase in buffers required for some purpose (egress traffic) will be offset by a reduction in buffers required for other purposes (transit traffic). Also, routers today typically deal with high-fanout channelized interfaces and with multi-VC ATM interfaces, implying that the requirement of buffering simultaneously arriving cells from multiple packets and sources is something that routers typically do today. This is not meant to imply that the required buffer size and performance is inexpensive, but rather is meant to observe that it is a solvable issue.

ATM equipment provides traffic shaping, in which the ATM cells associated with any one particular VC are intentionally not transmitted back to back, but rather are spread out over time in order to place less short term buffering load on switches. Since VC merge requires that all cells associated with a particular packet (or a particular AAL5 frame) are buffered before any cell from the packet can be transmitted, VC merge defeats much of the intent of traffic shaping. An advantage of VP merge is that it preserves traffic shaping through ATM switches acting as LSRs. While traffic shaping may generally be expected to reduce the buffering requirements in ATM switches (whether acting as MPLS switches or as native ATM switches), the precise effect of traffic shaping has not been studied in the context of MPLS.

4.3 Loop Handling

Generally, methods for dealing with loops can be split into three categories: Loop Survival makes use of methods which minimize the impact of loops, for example by limiting the amount of network resources which can be consumed by a loop; Loop Detection allows loops to be set up, but later detects these loops and eliminates them; Loop Prevention provides methods for avoiding setting up L2 switching in a way which results in a L2 loop.

Note that we are concerned here only with loops that occur in L2 switching. Transient loops at L3 will continue to be part of the normal IP operation, and will be handled the way that IP has been handling loops for years (see section 3.5).

Loop Survival:

Loop Survival refers to methods that are used to allow the network to operate well even though short term transient loops may be formed by the routing protocol. The basic approach to loop

Callon et al.   Expires March 2000   [Page 43]
□
IETF Draft   A Framework for MPLS   September 1999

survival is to limit the amount of network resources which are consumed by looping packets, and to minimize the effect on other (non-looping) traffic. Note that loop survival is the method used by conventional IP forwarding, and is therefore based on long and relatively successful experience in the Internet.

The most basic method for loop survival is based on the use to a TTL (Time To Live) field. The TTL field is decremented at each hop. If the TTL field reaches zero, then the packet is discarded. This method works well over those media which has a TTL field. This explicitly includes L3 IP forwarding. Also, assuming that the core MPLS specifications will include definition of a "shim" MPLS header for use over those media which do not have their own labels, in order to carry labels for use in forwarding of user data, the shim header will also include a TTL field.

However, there is considerable interest in using MPLS over L2 protocols which provide their own labels, with the L2 label used for MPLS forwarding. Specific L2 protocols which offer a label for this purpose include ATM and Frame Relay. However, neither ATM nor Frame Relay have a TTL field. This implies that this method cannot be used when basic ATM or Frame Relay forwarding is being used.

Another basic method for loop survival is the use of dynamic routing protocols which converge rapidly to non-looping paths. In some instances it is possible that congestion caused by looping data could affect the convergence of the routing protocol (see section 3.5). MPLS should be designed to prevent this problem from occurring. Given that MPLS uses the same routing protocols as are used for IP, this method does not need to be discussed further in this framework document.

Another possible tool for loop survival is the use of fair queuing. This allows unrelated flows of user data to be placed in different queues. This helps to ensure that a node which is overloaded with looping user data can nonetheless forward unrelated non-looping data, thereby minimizing the effect that looping data has on other data. We cannot assume that fair queuing will always be available. In practice, many fair queuing implementations merge multiple streams into one queue (implying that the number of queues used is less than the number of user data flows which are present in the network). This implies that any data which happens to be in the same queue with looping data may be adversely effected.

Loop Detection:

Loop Detection refers to methods whereby a loop may be set up at L2, but the loop is subsequently detected. When the loop is detected, it may be broken at L2 by dropping the label

Callon et al.   Expires March 2000   [Page 44]
▯
IETF Draft   A Framework for MPLS   September 1999

relationship, implying that packets for a set of destinations must be forwarded at L3.

A possible method for loop detection is based on transmitting a "loop detection" control packet (LDCP) along the path towards a specified destination whenever the route to the destination changes. This LDCP is forwarded in the direction that the label specifies, with the labels swapped to the correct next hop value. However, normal L2 switching cannot be used because each hop needs to examine the packet to check for loops. The LDCP is forwarded towards that destination until one of the following happens: (i) The LDCP reaches the last MPLS node along the path (ie the next hop is either a router which is not participating in MPLS, or is

the final destination host); (ii) The TTL of the LDCP expires
(assuming that the control packet uses a TTL, which is optional
but not absolutely necessary), or (iii) The LDCP returns to the
node which originally transmitted it. If the latter occurs, then
the packet has looped and the node which originally transmitted
the LDCP stops using the associated label, and instead uses L3
forwarding for the associated destination addresses. One problem
with this method is that once a loop is detected it is not known
when the loop clears. One option would be to set a timer, and to
transmit a new LDCP when the timer expires.

Loop detection may also be achieved via a Path Vector control
message. A Path Vector contains a list of the LSRs that label
distribution Control message has traversed. Each LSR which
propagates a control packet to either create or modify an LSP adds
its own unique identifier to the Path Vector list. An LSR that
receives a message with a Path Vector that contains its own
identifier detects that the message has traversed a loop.

An alternate method counts the hops to each egress node, based on
the routes currently available. Each node advertises its distance
(in hop counts) to each destination. An egress node advertises the
destinations that it can reach directly with an associated hop
count of one. For each destination, a node computes the hop count
to that destination based on adding one to the hop count
advertised by its actual next hop used for that destination. When
the hop count for a particular destination changes, the hop counts
needs to be readvertised.

In addition, the first of the loop prevention schemes discussed
below may be modified to provide loop detection.

Loop Prevention:

Loop prevention makes use of methods to ensure that loops are
never set up at L2. This implies that the labels are not used
until some method is used to ensure that following the label

Callon et al.   Expires March 2000   [Page 45]
□
IETF Draft   A Framework for MPLS   September 1999

towards the destination, with associated label swaps at each
switch, will not result in a loop. Until the L2 path (making use
of assigned labels) is available, packets are forwarded at L3.

Loop prevention requires explicit signaling of some sort to be
used when setting up an L2 stream.

One method of loop prevention requires that labels be propagated
starting at the egress switch. The egress switch signals to
neighboring switches the label to use for a particular
destination. That switch then signals an associated label to its
neighbors, etc. The control packets which propagate the labels
also include the path to the egress (as a list of routerIDs). Any
looping control packet can therefore be detected and the path not
set up to or past the looping point.

During routing changes, a diffusion mechanism may be used to
prevent the formation of L2 loops. The purpose of the diffusion
computation is to prune the tree of an LSR that has detected a
route change for a given FEC, such that all upstream LSR's from
the tree that would be on a looping path are removed. It is only

after those LSR's are removed from the tree that it is safe to
replace the old LSP with the new LSP (and the old LSP can be
released).

The diffusion mechanism is an extension of the Path Vector
mechanism. An LSR, D, that detects that the next hop for an FEC
has changed, transmits a query message with a Path Vector
containing its unique identifier to its upstream neighbors. An
LSR, U, that receives such a query will determine if D is the next
hop for the given FEC. If not, then U may return "OK", meaning
that as far as node U is concerned it is safe for node D to switch
over to the new LSP. If node D is the next hop, then node U checks
the Path Vector to see if its unique identifier is already
present. If so, then a route loop is detected; in this case, node
U responds with a "LOOP" message, and node D will prune node U off
of its tree. If no loop is detected, then node U adds its unique
identifier to the Path Vector, and propagates the query message to
each of its upstream neighbors. The diffusion computation
continues to propagate upstream along each of the paths in the
tree until an ingress or looping LSR is found. Once an LSR has
received a response from each of its upstream neighbors, it may
then return an "OK" message to its downstream neighbor. When the
original node, node D, receives a response from each of its
neighbors, it is safe to replace the old LSP with the new one
because all the paths that would have looped have been pruned from
the tree.

An alternative method of loop prevention is the "colored"
mechanism. The heart of the Colored Thread (CT) algorithm

Callon et al.   Expires March 2000   [Page 46]
□
IETF Draft   A Framework for MPLS   September 1999

propagates a procedure that gives a color to each link along the
LSP in the downstream direction. The color is composed of two
fixed-length objects; the address of the node that created the
color and a local identifier that is unique within the creating
node. A loop-free LSP is established when the node that triggered
the coloring procedure receives an acknowledgment for the
procedure from its downstream node. During the coloring procedure,
a set of attributes (color, hop count, TTL), referred to as a
thread, is propagated downstream. A node that finds a change in
the next hop creates a color and passes it on the outgoing link to
the new next hop. If a node receives a color on an incoming link,
it either (a) passes the received color or (b) creates a new color
and passes it, on the outgoing link to the next hop. The coloring
procedure is propagated downstream until the LSP turns out to be
loop-free or a loop is found. In case (i), a positive
acknowledgment (ACK) is returned hop-by-hop to upstream nodes. In
case (ii), the coloring procedure is stalled and no ACK is
returned. [LOOP-COLOR]

Another option is to use explicit routing to set up label
bindings. This precludes the possibility of looping, since the
entire path is computed by one node. This also allows non-looping
paths to be set up provided that the edge switch has a view of the
topology which is reasonably close to reality (if there are
operational links which the edge switch doesn't know about, it
will simply pick a path which doesn't use those links; if there
are links which have failed but which the edge switch thinks are
operational, then there is some chance that the setup attempt will
fail but in this case the attempt can be retried on a separate

path). Note therefore that non-looping paths can be set up with
this method in many cases where distributed routing plus hop by
hop forwarding would not actually result in non-looping paths.
This method is similar to the method used by standard ATM routing
to ensure that SVCs are non-looping [PNNI].

Explicit routing is applicable if it is configured, or if the
routing protocol gives the edge switch sufficient information to
set up the explicit route, implying that the protocol must be
either a link state protocol (such as OSPF) or a path vector
protocol (such as BGP). This method also requires some overhead
for the call setup before label-based forwarding can be used. If
the network topology changes in a manner which breaks the existing
path, then a new path will need to be explicit routed from the
edge switch. Due to this overhead this method is probably only
appropriate if other significant advantages are also going to be
obtained from having a single node (the edge switch) coordinate
the paths to be used.

If label distribution is piggybacked on the routing protocol (see
section 4.1.2), then loop prevention is only possible if the

Callon et al.   Expires March 2000   [Page 47]
□
IETF Draft   A Framework for MPLS   September 1999

routing protocol itself does loop prevention.

What To Do If A Loop Is Detected:

With all of these schemes, if a loop is known to exist then the L2
label-swapped path is not set up. This leads to the obvious
question of what does an MPLS node do when it doesn't have a label
for a particular destination, and a packet for that destination
arrives to be forwarded? If possible, the packet is forwarded
using normal L3 (IP) forwarding. There are two issues that this
raises: (i) What about nodes which are not capable of L3
forwarding; (ii) Given the relative speeds of L2 and L3
forwarding, does this work?

Nodes which are not capable of L3 forwarding obviously can't
forward a packet unless it arrives with a label, and the
associated next hop label has been assigned. Such nodes, when they
receive a packet for which the next hop label has not been
assigned, must discard the packet. It is probably safe to assume
that if a node cannot forward an L3 packet, then it is probably
also incapable of forwarding an ICMP error report that it
originates. This implies that the packet will need to be discarded
in this case.

In many cases L2 switching will be significantly faster than L3
forwarding (allowing faster forwarding is a significant motivation
behind the work on MPLS). This implies that if a node is
forwarding a large volume of traffic at L2, and a change in the
routing protocol causes the associated labels to be lost
(necessitating L3 forwarding), in some cases the node will not be
capable of forwarding the same volume of traffic at L3. This will
of course require that packets be discarded. However, in some
cases only a relatively small volume of traffic will need to be
forwarded at L3. Thus forwarding at L3 when L2 is not available is
not necessarily always a problem. There may be some nodes which
are capable of forwarding equally fast at L2 and L3 (for example,
such nodes may contain IP forwarding hardware which is not

available in all nodes). Finally, when packets are lost this will cause TCP to backoff, which will in turn reduce the load on the network and allow the network to stabilize even at reduced forwarding rates until such time as the label bindings can be reestablished.

In many cases MPLS may be used for traffic engineering. In these cases failure of an LSP may cause packets which would have taken that LSP to be forwarded (using L3 forwarding) along paths which are not consistent with the traffic engineering solution. This could in turn cause congestion. In these cases packets may need to be discarded even if the LSRs are capable of full line rate L3 forwarding. This may cause problems very similar to those

Callon et al.   Expires March 2000   [Page 48]
□
IETF Draft   A Framework for MPLS   September 1999

discussed in the previous paragraph.

Note that in most cases loops will be caused either by configuration errors, or due to short term transient problems caused by the failure of a link. If only one link goes down, and if routing creates a normal "tree-shaped" set of paths to any one destination, then the failure of one link somewhere in the network will effect only one link's worth of data passing through any one node in the network. This implies that if a node is capable of forwarding one link's worth of data at L3, then in many or most cases it will have sufficient L3 bandwidth to handle looping data.

4.4 Interoperation with NHRP

When label switching is used over ATM, and there exists an LSR which is also operating as a Next Hop Client (NHC), the possibility of direct interaction arises. That is, could one switch cells between the two technologies without reassembly? To enable this several important issues must be addressed.

The encapsulation must be acceptable to both MPLS and NHRP. If only a single label is used, then the null encapsulation could be used. Other solutions could be developed to handle label stacks.

NHRP must understand and respect the granularity of a stream.

Currently NHRP resolves an IP address to an ATM address. The response may include a mask indicating a range of addresses. However, any VC to the ATM address is considered to be a viable means of packet delivery. Suppose that an NHC NHRPs for IP address A and gets back ATM address 1 and sets up a VC to address 1. Later the same NHC NHRPs for a totally unrelated IP address B and gets back the same ATM address 1. In this case normal NHRP behavior allows the NHC to use the VC (that was set up for destination A) for traffic to B [RFC2332].

Note: In this section we will refer to a VC set up as a result of an NHRP query/response as a shortcut VC.

If one expects to be able to label switch the packets being received from a shortcut VC, then the label switch needs to be informed as to exactly what traffic will arrive on that VC and that mapping cannot change without notice. Currently there exists no mechanism in the defined signaling of an shortcut VC. Several means are possible. A binding, equivalent to the binding in LDP,

could be sent in the setup message. Alternatively, the binding of
prefix to label could remain in an LDP session (or whatever means
of label distribution as appropriate) and the setup could carry a
binding of the label to the VC. This would leave the binding
mechanism for shortcut VCs independent of the label distribution

Callon et al.   Expires March 2000   [Page 49]
□
IETF Draft   A Framework for MPLS   September 1999

mechanism.

A further architectural challenge exists in that label switching
is inherently unidirectional whereas ATM is bi-directional. The
above binding semantics are fairly straight-forward. However,
effectively using the reverse direction of a VC presents further
challenges.

Label switching must also respect the granularity of the shortcut
VC. Without VC merge, this means a single label switched flow must
map to a VC. In the case of VC merge, multiple label switched
streams could be merged onto a single shortcut VC. But given the
asymmetry involved, there is perhaps little practical use.

Another issue is one of practicality and usefulness. What is sent
over the VC must be at a fine enough granularity to be label
switched through receiving domain. One potential place where the
two technologies might come into play is in moving data from one
campus via the wide-area to another campus. In such a scenario,
the two technologies would border precisely at the point where
summarization is likely to occur. Each campus would have a
detailed understanding of itself, but not of the other campus. The
wide-area is likely to have summarized knowledge only. But at such
a point level 3 processing becomes the likely solution.

4.5. Operation in a hierarchy

MPLS allows hierarchical operation, through use of a label stack.
This allows MPLS to simultaneously be used for routing at a fine
grain level (for example, between individual routers within an
ISP) and at a higher "area by area" or "domain by domain" level.

4.5.1 Example of Hierarchical Operation

Figure 1 illustrates an example of how MPLS may operate in a
hierarchy. This example illustrates three transit routing domains
(Domain #1, #2, and #3). For example, these three domains may
represent internet service providers. Domain Boundary Routers are
illustrated in each domain (routers R1 and R2 in domain #1,
routers R3 and R8 in domain #2, and routers R9 and R10 in domain
#3. Suppose that these domain boundary routers are operating BGP.

Internal routers are not illustrated in domains 1 and 3. However,
internal routers are illustrated within domain #2. In particular,
the path between routers R3 and R8 follows the internal routers
R4, R5, R6, and R7 within domain #2.

IETF Draft   A Framework for MPLS   September 1999


```
 .................      ...........................      .................
 .              .      .                         .      .               .
 .              .      .     .                   .      .               .
 .R1            .   R2------R3                 R8------R9          R10.
 .              .      . .  \                 /  .      .               .
 .              .      . .   R4---R5---R6---R7   .      .               .
 .              .      .                         .      .               .
 .  Domain#1    .      .         Domain#2        .      .   Domain#3    .
 .................      ...........................      .................
```
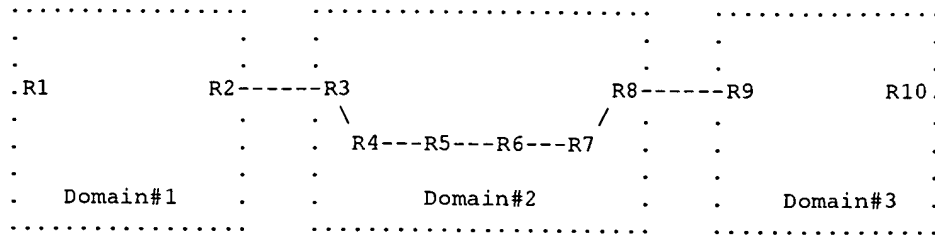
                Figure 1: Example of the Use of MPLS in a Hierarchy

In this example there are two levels of routing taking place. For
example, OSPF may be used for routing within Domain #2. In this
case the routers R3, R4, R5, R6, R7, and R8 may be running OSPF
amongst themselves in order to compute routes within Domain #2.
The domain boundary routers (R1, R2, R3, R8, R9, and R10) operate
BGP in order to determine paths between routing domains.

MPLS allows label forwarding to be done independently at multiple
levels. In this example, MPLS may be used at the BGP level
(between routers R1, R2, R3, R8, R9, and R10) and at the OSPF
level (between routers R4, R5, R6, and R7). Thus when the IP
packet traverses Domain number 2, it will contain two labels,
encoded as a "label stack". The higher level label would be used
between routers R3 and R8. This would be encapsulated inside a
header specifying a lower level label used within domain 2.

Consider the forwarding operation that takes place at router R3.
In this case, R3 will receive a packet from R2 containing a single
label (the BGP level label). R3 will need to swap BGP level labels
in order to put the label that R8 expects. R3 will also need to
add an OSPF-level label, as is expected by R4. R3 therefore
"pushes down" the BGP level label in the label stack, by adding a
lower level label. Also note that the actual label switching
operation performed by R3 can be optimized to allow very simple
forwarding: R3 receives a single incoming label from R2, and can
map this label into the new label header to be prepended to the
packet, it just happens that the new label header to be added by
R3 contains two labels rather than one.

4.5.2 Components Required for Hierarchical Operation

   In order for MPLS to operate in a hierarchy, there are three
   things which must be accomplished:

   - Hierarchical Label Exchange in LDP
     The Label Distribution Protocol needs to exchange labels at
     each level of the hierarchy. In our example, R3 needs to
     exchange label bindings with R8 for operation at the BGP

IETF Draft   A Framework for MPLS   September 1999

     level. At the same time, R3 needs to exchange label

bindings with R4 (and R4 needs to exchange label bindings
with R5) for operation at the OSPF level. The control
component for hierarchical labeling is essentially the same
as that for single level tagging, except that labels are
exchanged not just among physically adjacent LSRs but
between those switching on the same level in the tag stack.

- Label Stack
  Multiple labels need to be carried in data packets. For
  example, when a data packet is being carried across domain
  #2, the data packet needs to be encapsulated in a header
  which carries BGP level label, and the resulting packet
  needs to be carried in a header which carries an OSPF level
  label.

- Configuration
  It is necessary for routers to know when hierarchical label
  switching is being used.

4.5.3 Some Restrictions on Use of Hierarchical MPLS

Consider the example in figure 1. In this case, the BGP-level
label is encoded by router R1. Label switching is employed for
packet forwarding at R2, R3, R8, and R9. This is only possible if
R1 knows the right label to use, implying that the granularity
used in mapping packets to forwarding equivalence classes is the
same at routers R2, R3, R8, and R9.

We can consider some specific examples to illustrate the issue:

Suppose that the destination host is within domain 3. In this
case, it is very likely that router R9 will forward the packet
based on a finer grain than was used previously. For example, a
relatively short address prefix may be used for advertising the
addresses reachable in domain 3, while longer (more specific)
address prefixes may be used for specific areas or subnets within
domain 3. In this case router R1 may assign a BGP level label to
the packet, and label based forwarding at the BGP level may be
used by routers R1, R2, R3, and R8. However, router R9 will need
to make use of layer 3 forwarding.

Alternatively, suppose that domain 3 is an Internet Service
Provider, which offers service to multiple routing domains.
Suppose that in this case domain 3 makes use of a single CIDR
address block (based on a single address prefix), with smaller
address blocks (corresponding to longer address prefixes) assigned
to each of multiple domains who get their Internet service from
domain 3. Suppose that the destination for a particular IP packet
is contained in one of these smaller domains whose addresses are

Callon et al.   Expires March 2000   [Page 52]
□
IETF Draft   A Framework for MPLS   September 1999

contained in the larger address block assigned to and administered
by domain 3. Again in this case router R9 will need to make use of
label based forwarding.

Let's consider another possible complication: Suppose that router
R1 is an MPLS node, but that some of the internal routers within
domain 1 do not know about MPLS. In this case, suppose that R1
encapsulates an IP packet in an MPLS header in order to carry the
BGP level label. In this case the non-MPLS-capable routers within

domain 1 will not know what to do with the MPLS header. This
implies that MPLS can be used at a higher level (such as between
the border routers R1 and R2 in our example) only if either the
lower level routers (such as the routers within domain 1)are also
using MPLS, or the MPLS header is itself encapsulated within an IP
header for transmission across the domain.

These examples imply that there are some cases where IP forwarding
will be required in a hierarchy. While hierarchical MPLS may be
useful in many cases, it does not replace layer 3 forwarding.

4.5.4 The Relationship between MPLS hierarchy and Routing Hierarchy

4.5.4.1 Stacked Labels in a Flat Routing Environment

The label stacking mechanism can be useful in some scenarios
independent of routing hierarchy.

The basic concept of stacking is to provide a mechanism to
segregate streams within a switched path. Under normal operation,
when packets are encapsulated into a single L2 header, if multiple
streams are forwarded into a switched path, it will require L3
processing to segregate a certain stream at the end of the
switched path. The stacking mechanism provides an easy way to
maintain the identity of various streams which are merged into a
single switched path.

One useful application of this technique is in Virtual Private
Networks. The packets can be switched both at the ingress and
egress nodes of the provider network. A packet coming in at one
end of a customer network contains an encapsulated header with the
VPN label. At the VPN ingress node, the header is "popped", to
provide the label for switching through the VPN. Further, this
header is then "pushed" with an encapsulation of the far end
customer label. At the VPN egress node, the packet header is
"popped" again, and the new header provides the label for
switching through the customer site. This enables one to provide
customers with benefits of VPN with end-to-end switching for
optimal performance.

Another interesting use can be in conjunction with RSVP flows. In

Callon et al.   Expires March 2000   [Page 53]
☐
IETF Draft   A Framework for MPLS   September 1999

RSVP, senders flows can be logically merged under a single
resource reservation using the Shared and the Wildcard filters.
The stacking mechanism can be used to merge flows into a single
label and the shared QoS can be applied to the single label on top
of the stack. Since sender flows within the merged switched path
maintain their identity, it is easy to demerge at a downstream
node without requiring L3 processing of the packets. Another
similar application can be merging of several premium service
flows with similar QoS into a single switched path. This helps in
conserving labels in backbone of a large networks.

Yet another useful application can be DVMRP tunnels similar in
concept to the DVMRP tunnels used in the existing Mbone. The
ingress node to the DVMRP switched tunnels encapsulates the label
learned from the egress node of the DVMRP tunnel for a particular
(S,G) pair before forwarding packets into the DVMRP tunnel. The
egress node of the tunnel just pops the top label and switches the

packet based on the interior label.

Note that the use of tunnels can be also quite beneficial in a non-hierarchical environment. Take for example the case where a domain contains a subset of MPLS nodes. The MPLS egress can advertise labels for the routes which are within the domain, but are external to the MPLS core. The ingress node can encapsulate packets for these destinations within the header for the aggregated switched path that crosses the MPLS domain.

It is not evident if this technique has any useful application in a flat routing domain, but can be used in conjunction with explicit routing when providing specialized services. The multiple levels of encapsulation can also be used like loose source routing.

4.5.4.2 Flat labels in a Hierarchical Routing Environment

It is also possible in some environments to use a single level of label in a network using hierarchical routing. This is for example possible in the case of a two level OSPF network in which the primary purpose of the network is to support external routes. Specifically, (depending upon the types of area hierarchy used) OSPF allows external routes to be advertised throughout an OSPF routing domain, with each external route associated with the routerID of the router with reachability to the specific route. This implies that it is possible to set up an LSP to every router in the routing domain, and then use the LSP for packets destined to the associated external routes.

4.5.4.3 Configuration of the Hierarchy

The possibility of having a variety of different relationships

Callon et al.   Expires March 2000   [Page 54]
□
IETF Draft   A Framework for MPLS   September 1999

between the routing hierarchy and the MPLS hierarchy leads to an obvious question: How is the relationship between the two hierarchies to be determined? At first glance it would seem that this generality leads to a relatively complex configuration issue, and it could be difficult to ensure consistent configuration of the network.

One possible solution is to have the MPLS hierarchy default to using the same hierarchy structure as is used for routing, with each area and domain boundary (as used by routing) also implying an MPLS domain boundary. This would allow the normal default operation to conform to the type of operation that we might expect to be used in most situations, and would allow a common means of interoperation which we would expect all vendors of MPLS compliant equipment to support.

4.5.5 Some Advantages of Hierarchical MPLS

The use of hierarchical MPLS allows the routers internal to a transit routing domain to be isolated from the BGP-level routing information. In our example network, routers R4, R5, R6, and R7 can forward packets based solely on the lower level label. These internal routers do not need to know anything at all about higher level IP routing. Note that this advantage is not available in conventional IP forwarding: If the internal routers within a

routing domain forward IP packets based on the destination IP
address, then the internal routers need to know which route to use
for any particular destination IP address. By combining
hierarchical routing with label stacks MPLS is able to decouple
the exterior and interior protocols. MPLS switches within a domain
(interior switches) need only carry the reachability information
for nodes in the domain. The MPLS border switches for the domain
still, of course, carry the external routes.

Use of hierarchical MPLS also extends the simpler forwarding
offered by MPLS to domain boundary routers.

MPLS places no bound on the number of labels that may be present
in a label stack. In principle this means that MPLS can support
multiple levels of routing hierarchy.

4.6 Interoperation of MPLS systems with "Conventional" ATM

If we consider the implementation of MPLS on ATM switches we can
imagine several possibilities.

We might remove ATM Forum control plane completely. This is the
approach taken by Ipsilon in their IP Switching approach, and
allows ATM switches to operate as MPLS LSRs.

Callon et al.   Expires March 2000   [Page 55]
□
IETF Draft   A Framework for MPLS   September 1999

Alternately, we could build a system that supports a "Ships in the
night" (SIN) mode of operation where the ATM Forum and MPLS
control planes both run on the same hardware but are isolated from
each other, ie, they do not interact. This allows a single device
to simultaneously operate as both an MPLS LSR and an ATM switch.

We feel that the MPLS architecture should allow both of these
models. We note, however, that neither of them addresses the issue
of operation of MPLS over a public ATM network, ie over a network
that supports tariffed access to PVCs and ATM Forum SVCs. Because
public ATM service exists and will, presumably, become more
pervasive in the future we feel that another model needs to be
included in the architecture and be supported by MPLS. We call
this model the "integrated" model. In essence it is the same as
the SIN model but without the restriction that the two control
planes are isolated. In the integrated model the MPLS control
plane is able to use the ATM control plane to setup SVCs as
needed. An example of this integrated model that allows the
coexistence and interoperation between ATM and MPLS is the CSR
proposal from Toshiba.

Note that there is a distinction relevant to the protocol
specification process between the SIN and the Integrated approach.
SIN does not require specification other than to require that it
be transparent to both the MPLS and ATM control planes (ie neither
should know of the others existence). Realization of SIN on a
particular machine is purely an engineering challenge for the
implementers. The Integrated model on the other hand requires
specification of procedures for the use of SVCs and association of
labels with them.

4.7 Multicast

This section is FFS.

4.8 Multipath

Many IP routing protocols support the notion of equal-cost multipath routes, in which a router maintains multiple next hops for one destination prefix when two or more equal-cost paths to the prefix exist. There are a few possible approaches for handling multipath with MPLS.

In this discussion we will use the term "multipath node" to mean a node which is keeping track of multiple switched paths from itself for a single destination.

The first approach maintains a separate switched path from each ingress node via one or more multipath nodes to a merge point. This requires MPLS to distinguish the separate switched paths, so

Callon et al.   Expires March 2000   [Page 56]
□
IETF Draft   A Framework for MPLS   September 1999

that learning of a new switched path is not misinterpreted as a replacement of the same switched path. This also requires an ingress MPLS node be capable of distributing the traffic among the multiple switched paths. This approach preserves switching performance, but at a cost of proliferating the number of switched paths. For example, each switched path consumes a distinct label.

The second approach establishes only one switched path from any one ingress node to a destination. However, when the paths from two different ingress nodes happen to arrive at the same node, that node may use different paths for each (implying that the node becomes a multipath node). Thus the switched path chosen by the multipath node may assign a different downstream path to each incoming stream. This conserves switched paths and maintains switching performance, but cannot balance loads across downstream links as well as the other approaches, even if switched paths are selectively assigned. An issue with this approach is that the L2 path may be different from the normal L3 path, as traffic that otherwise would have taken multiple distinct paths is forced onto a single path.

The third approach allows a single stream arriving at a multipath node to be split into multiple streams, by using L3 forwarding at the multipath node. For example, the multipath node might choose to use a hash function on the source and destination IP addresses, in order to avoid misordering packets between any one IP source and destination. This approach conserves switched paths at the cost of switching performance.

4.9 Host Interactions

There are a range of options for host interaction with MPLS:

The most straightforward approach is no host involvement. Thus host operation may be completely independent of MPLS, rather hosts operate according to other IP standards. If there is no host involvement then this implies that the first hop requires an L3 lookup.

If the host is ATM attached and doing NHRP, then this would allow the host to set up a Virtual Circuit to a router. However this

brings up a range of issues as was discussed in section 4.4
("interoperation with NHRP").

On the ingress side, it is reasonable to consider having the first
hop LSR provide labels to the hosts, and thus have hosts attach
labels for packets that they transmit. This could allow the first
hop LSR to avoid an L3 lookup. It is reasonable here to have the
host request labels only when needed, rather than require the host
to remember all labels assigned for use in the network.

Callon et al.   Expires March 2000   [Page 57]
□
IETF Draft   A Framework for MPLS   September 1999 ;

On the egress side, it is questionable whether hosts should be
involved. For scaling reasons, it would be undesirable to use a
different label for reaching each host.

4.10 Explicit Routing

There are two options for Route Selection: (1) Hop by hop routing,
and (2) Explicit routing.

An explicitly routed LSP is an LSP where, at a given LSR, the LSP
next hop is not chosen by each local node, but rather is chosen by
a single node (usually the ingress or egress node of the LSP). The
sequence of LSRs followed by an explicit routing LSP may be chosen
by configuration, or by an algorithm performed by a single node
(for example, the egress node may make use of the topological
information learned from a link state database in order to compute
the entire path for the tree ending at that egress node).

With MPLS the explicit route needs to be specified at the time
that Labels are assigned, but the explicit route does not have to
be specified with each L3 packet. This implies that explicit
routing with MPLS is relatively efficient (when compared with the
efficiency of explicit routing for pure datagrams).

Explicit routing may be useful for a number of purposes such as
allowing policy routing and/or facilitating traffic engineering.

4.10.1 Establishment of Point to Point Explicitly Routed LSPs

In order to establish a point to point explicitly routed LSP, the
signaling messages used to set up the LSP must contain the
explicit route. This implies that the LSP is set up in order
either from the ingress to the egress, or from the egress to the
ingress.

One node needs to pick the explicit route: This may be done in at
least two possible ways: (i) by configuration (eg, the explicit
route may be chosen by an operator, or by a centralized server of
some kind); (ii) By use of a routing protocol which allows the
ingress and/or egress node to know the entire route to be
followed. This would imply the use of a link state routing
protocol (in which all nodes know the full topology) or of a path
vector routing protocol (in which the ingress node is told the
path as part of the normal operation of the routing protocol).

Note: The normal operation of path vector routing protocols (such
as BGP) does not provide the full set of routers along the path.
This implies that either a partial source route only would be

provided (implying that LSP setup would use a combination of hop

by hop and explicit routing), or it would be necessary to augment the protocol in order to provide the complete explicit route.

In the point to point case, it is relatively straightforward to specify the route to use: This is indicated by providing the addresses of each LSR on the LSP.

4.10.2 Explicit and Hop by Hop routing: Avoiding Loops

In general, an LSP will be explicit routed specifically because there is a good reason to use an alternative to the hop by hop routed path. This implies that the explicit route is likely to follow a path which is inconsistent with the path followed by hop by hop routing. If some of the nodes along the path follow an explicit route but some of the nodes make use of hop by hop routing (and ignore the explicit route), then inconsistent routing may result and in some cases loops (or severely inefficient paths) may form. This implies that for any one LSP, there are two possible options: (i) The entire LSP may be hop by hop routed; or (ii) The entire LSP may be explicit routed.

For this reason, it is important that if an explicit route is specified for setting up an LSP, then that route must be followed in setting up the LSP.

There is a related issue when a link or node in the middle of an explicitly routed LSP breaks: In this case, the last operating node on the upstream part of the LSP will continue receiving packets, but will not be able to forward them along the explicitly routed LSP (since its next hop is no longer functioning). In this case, it is not in general safe for this node to forward the packets using L3 forwarding with hop by hop routing (unless loose source routing is present). Instead, the packets must be discarded, and the upstream partition of the explicitly routed LSP must be torn down.

Where part of an Explicitly Routed LSP breaks, the node which originated the LSP needs to be told about this. For robustness reasons the MPLS protocol design should not assume that the routing protocol will tell the node which originated the LSP. For example, it is possible that a link may go down and come back up quickly enough that the routing protocol never declares the link down. Rather, an explicit MPLS mechanism is needed.

4.10.3 Merge and Explicit Routing

Explicit Routing is slightly more complex with a multipoint to point LSP (ie, in the case that stream merge is used).

In this case, it is not possible to specify the route for the LSP

as a simple list of LSRs (since the LSP does not consist of a
simple sequence of LSRs). Rather the explicit route must specify a
tree. There are several ways that this may be accomplished.
Details are outside the scope of this document.

4.10.4 Using Explicit Routing for Traffic Engineering

In the Internet today it is relatively common for ISPs to make use
of a Frame Relay or ATM core, which interconnects a number of IP
routers. The primary reason for use of a switching (L2) core is to
make use of low cost equipment which provides very high speed
forwarding. However, there is another very important reason for
the use of a L2 core: In order to allow for Traffic Engineering.

Traffic Engineering (also known as bandwidth management) refers to
the process of managing the routes followed by user data traffic
in a network in order to provide relatively equal and efficient
loading of the resources in the network (ie, to ensure that the
bandwidth on links and nodes are within the capabilities of the
links and nodes).

Some rudimentary level of traffic engineering can be accomplished
with pure datagram routing and forwarding by adjusting the metrics
assigned to links. For example, suppose that there is a given link
in a network which tends to be overloaded on a long term basis.
One option would be to manually configure an increased metric
value for this link, in the hopes of moving some traffic onto
alternate routes. This provides a rather crude method of traffic
engineering and provides only limited results.

Another method of traffic engineering is to manually configure
multiple PVCs across a L2 core, and to adjust the route followed
by each PVC in an attempt to equalize the load on different parts
of the network. Where necessary, multiple PVCs may be configured
between the same two nodes, in order to allow traffic to be split
between different paths. In some topologies it is much easier to
achieve efficient non-overlapping or minimally-overlapping paths
via this method (with manually configured paths) than it would be
with pure datagram forwarding. A similar ability can be achieved
with MPLS via the use of manual configuration of the paths taken
by LSPs.

A related issue is the decision on where merge is to occur. Note
that once two streams merge into one stream (forwarded by a single
label) then they cannot diverge again at that level of the MPLS
hierarchy (ie, they cannot be bifurcated without looking at a
higher level label or the IP header). Thus there may be times when
it is desirable to explicitly NOT merge two streams even though
they are to the same egress node and FEC. Non-merge may be
appropriate either because the streams will want to diverge later

in the path (for example, to avoid overloading a particular
downstream link), or because the streams may want to use different
physical links in the case where multiple slower physical links
are being aggregated into a single logical link for the purpose of
IP routing.

As a network grows to a very large size (on the order of hundreds
of LSRs), it becomes increasingly difficult to handle the
assignment of all routes via manual configuration. However,
explicit routing allows several alternatives:

1. Partial Configuration: One option is to use
   automatic/dynamic routing for most of the paths through
   the network, but then manually configure some routes. For
   example, suppose that full dynamic routing would result in
   a particular link being overloaded. One of the LSPs which
   uses that link could be selected and manually routed to
   use a different path.

2. Central Computation: One option would be to provide long
   term network usage information to a single central
   management facility. That facility could then run a global
   optimization to compute a set of paths to use. Network
   management commands can be used to configure LSRs with the
   correct routes to use.

3. Egress Computation: An egress node can run a computation
   which optimizes the path followed for traffic to itself.
   This cannot of course optimize the entire traffic load
   through the network, but can include optimization of
   traffic from multiple ingress's to one egress. The reason
   for optimizing traffic to a single egress, rather than
   from a single ingress, relates to the issue of when to
   merge: An ingress can never merge the traffic from itself
   to different egresses, but an egress can if desired chose
   to merge the traffic from multiple ingress's to itself.

4.11 TTL and Traceroute

   Traceroute is a useful method which is widely used for management
   of IP networks. It is therefore highly desirable for traceroute
   and TTL to be preserved in networks where MPLS is used. TTL can
   also be useful to minimize the impact of loops (ie, as an aid to
   loop survival).

   In cases where the MPLS shim header is used, and where the IP
   packets are normal Internet packets (ie, not part of a VPN), TTL
   can optionally be handled in a way which is semantically identical
   to operation in native IP networks. The ingress node, when
   encapsulating an IP packet in the MPLS shim header, copies the TTL

□
IETF Draft   A Framework for MPLS   September 1999

   from the IP header to the MPLS Shim Header. LSRs decrement the
   TTL, and behave as normal IP routers in the case that the TTL
   reaches zero (ie, discard the IP packet and return an ICMP error
   report). Egress routers copy the TTL from the MPLS shim header
   back to the IP header.

   Where multiple MPLS shim headers are used in a label stack, TTL
   can be handled in essentially the same manner. When a LSR pushes a
   new header onto the stack, the TTL is copied from the previous
   shim header to the new header. When an LSR pops a header off of
   the stack, TTL is copied in the other direction.

   Some carriers may choose to avoid exposing the topology (or even
   the diameter) of their networks to customers. One way to do this

is to treat an entire LSP crossing the carrier network as a single hop from the point of view of IP forwarding. In this case the ingress router places a value in the TTL field of the shim header which is independent of the TTL value found from the IP header. Similarly the decapsulating router strips off the MPLS header and forwards based on the IP header, but does not copy TTL values. Routers which are in the middle of the LSP (neither ingress nor egress) decrement the TTL contained in the MPLS shim header, but do not return an error report if the TTL is expired.

There is a problem with the handling of ICMP error reports when VPNs are supported using MPLS. In this case, the IP address space used in the IP packet (carried over the LSP) might be local to the VPN, and therefore might not be understood by the LSR which detects that the TTL has reached zero. In addition, core LSRs might not necessarily know which LSPs are supporting VPN traffic and which are supporting Internet traffic. For this reason in networks where VPNs are supported over MPLS, special precautions are needed. If the ingress node knows the path of the LSP, then it may discard the packet and return an ICMP error report (to the VPNs space) if the TTL is less than the length of the LSP. Alternatively, the TTL value used in the MPLS header may be independent of the TTL value in the IP header, and the entire LSP may be treated as a single hop from the perspective of datagram IP forwarding. Alternatively, ICMP error reports could be turned off in such networks.

One other potential solution to the ICMP error reporting problem is to use "bi-directional" LSPs. In this case, two LSPs may be created with the same endpoints, but which carry packets in opposite directions. These two LSPs are logically coupled together; that is, one LSP carries traffic from an originating node to a destination node, while the other carries traffic from the destination node to the originating node[TRAFENG]. When a packet has to be discarded that had been flowing on the LSP in one direction, the error report can be returned on the matching LSP in

Callon et al.   Expires March 2000   [Page 62]
□
IETF Draft   A Framework for MPLS   September 1999

the other direction. This is true even when the IP address space encapsulated inside the LSP is one which the LSR does not otherwise understand.

MPLS may also be used over L2 technologies which do not have TTL values (specifically ATM and Frame Relay). In this case, TTL and Traceroute may still be supported in some specific situations.

In our discussion we will assume that the MPLS encapsulation for operation of MPLS over ATM and Frame Relay media always use a shim header. Thus the packet would consist of an IP packet encapsulated inside an MPLS shim header, which would in turn be encapsulated for transmission over ATM or Frame Relay (eg, the IP packet and MPLS shim header may be encapsulated in an AAL5 frame, which would in turn be encapsulated inside ATM cells). If the shim header is not used, when manipulations of the TTL in the shim header as described below would be replaced by manipulations of the TTL inside the IP header.

The most straightforward case is one where ATM or Frame Relay is used for the entire path of the LSP, and where the ingress LSR knows the entire path of the LSP (for example, this may occur when

the LSP is set up based on complete source routing). In this case
the ingress router decrements the TTL by the length of the LSP. If
the TTL reaches zero or a negative number, then the IP packet is
discarded and an ICMP error report is returned by the ingress
router, but with a source address which indicates the node at
which the TTL would have expired. In this case in principle the
TTL which is decremented could be either the one in the IP header
or the one in the MPLS header. However, it allows more uniform
operation (compared to other situations) if the TTL in the shim
header is decremented by the ingress router by the length of the
path, and then the egress router copies the TTL from the MPLS
header into the IP packet.

In some cases the length of the LSP might be known, but not the
exact identity of the LSRs along the path (eg, the LSP is set up
via ordered control). In this case the TTL can be decremented as
above, but if the TTL would expire the packet could be forwarded
by some "out of band" (control processor to control processor)
path in order to get the packet to the LSR at which the TTL will
reach zero.

There may be cases where part of the LSP traverses ATM or Frame
Relay links (using an ATM or Frame Relay header), and part
traverses other media (using the shim header).

Some of the issues which come up in this situation are best
illustrated through use of an example. Suppose that in figure 2,
an LSP goes from R1 to R8. Thus R1 is the ingress LSR, and R8 is

Callon et al.   Expires March 2000   [Page 63]
□
IETF Draft   A Framework for MPLS   September 1999

                                      (

the egress LSR for this particular LSP. LSRs R3 and R6 have both
ATM interfaces and non-ATM interfaces. Thus the MPLS shim header
is used on the link from R1 to R2, and from R2 to R3. ATM is used
on the links from R3 to R4, R4 to R5, and R5 to R6. Finally, the
shim header is again used on the links from R6 to R7, and R7 to
R8.

```
    ...............................................................
    .                    .                .                .
    .                    .                .                .
    .R1------R2------R3            R6-----R7-----R8.
    .                  . \              /.                .
    .                  . R4------R5 .                .
    .                  .                .                .
    .  Shim Header    .      ATM      . Shim Header .
    ...............................................................
```
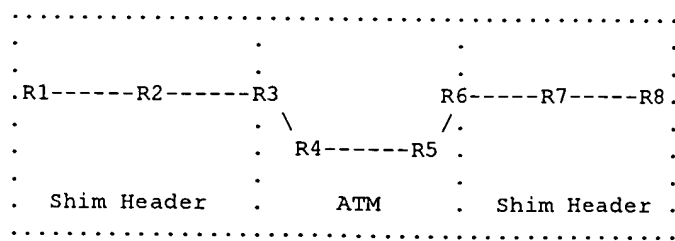
Figure 2: LSP spanning ATM and Shim Header Media

If egress-initiated ordered control is used, then it is possible
that when the LSP is first set up the signaling protocol could
keep track of the number of hops to the next LSR that will use a
shim header (and which therefore understands TTL). In our example
R3 could therefore know that it is three hops to R6 (which is the
next router which will use a shim header containing a TTL value).
R3 can therefore decrement the TTL by the appropriate value (3),
and return an error report if the TTL will expire.

If ingress-initiated ordered control or independent control is

used, then it is not clear how R3 will know the identity of the
next LSR which understands TTL (ie, will use a shim header instead
of an ATM or frame relay header). For example, suppose that
complete explicit routing with ingress control is used. In this
case R3 will know the complete path to the egress (R8), but will
not know which downstream links use ATM media and which uses the
shim header. Thus R3 will know that R6 is a downstream LSR for
this LSP, but will not know that R6 is the specific LSR which
removes the packet from the ATM media.

R6 will forward the packet based on the incoming label implicit in
the VPI/VCI from the ATM media, plus the existing shim header.
Thus the TTL used at this point will be based on that received in
the shim header. This implies that the TTL value in the shim
header needs to be valid, which in turn implies that R3 needs to
adjust the TTL value in the shim header to account for the length
of the path from R3 to R6.

4.12 LSP Control: Ordered versus Independent

Callon et al.   Expires March 2000   [Page 64]
囗
IETF Draft   A Framework for MPLS   September 1999

There is a choice to be made regarding whether the initial setup
of LSPs will be in an ordered mode, where the LSP is initiated by
the egress node, or independently by each individual node.

When LSP control is done independently, then each node may at any
time pass label bindings to its neighbors for each FEC recognized
by that node. In the normal case that the neighboring nodes
recognize the same FECs, then nodes may map incoming labels to
outgoing labels as part of the normal label switching forwarding
method.

When LSP control is done in an ordered manner, then the egress
node passes label bindings to its neighbors corresponding to any
FECs which leave the MPLS network at that egress node. Other nodes
must wait until they get a label from downstream for a particular
FEC before passing a corresponding label for the same FEC to
upstream nodes.

With independent control, since each LSR is independently
assigning labels to FECs, it is possible that different LSRs may
make inconsistent decisions. For example, an upstream LSR may make
a coarse decision (map multiple IP address prefixes to a single
label) while its downstream neighbor makes a finer grain decision
(map each individual IP address prefix to a separate label). With
downstream label assignment this can be corrected by having LSRs
withdraw labels that it has assigned which are inconsistent with
downstream labels, and replace them with new consistent label
assignments.

This may appear to be an advantage of ordered LSP control (since
with egress control the initial label assignments "bubble up" from
the egress to upstream nodes, and consistency is therefore easy to
ensure). However, even with ordered control it is possible that
the choice of egress node may change, or the egress may (based on
a change in configuration) change its mind in terms of the
granularity which is to be used. This implies the same mechanism
will be necessary to allow changes in granularity to bubble up to
upstream nodes. The choice of ordered or independent control may

therefore effect the frequency with which this mechanism is used, but will not effect the need for a mechanism to achieve consistency of label granularity.

Ordered control and independent control can interwork in a very straightforward manner: With either approach, (assuming downstream label assignment) the egress node will initially assign labels for particular FECs and will pass these labels to its neighbors. With either approach these label assignments will bubble upstream, with the upstream nodes choosing labels that are consistent with the labels that they receive from downstream.

Callon et al.   Expires March 2000   [Page 65]

IETF Draft   A Framework for MPLS   September 1999

The difference between the two techniques therefore becomes a tradeoff between avoiding a short period of initial thrashing on startup (in the sense of avoiding the need to withdraw inconsistent labels which may have been assigned using local control) versus the imposition of a short delay on initial startup (while waiting for the initial label assignments to bubble up from downstream). The protocol mechanisms which need to be defined are the same in either case, and the steady state operation is the same in either case.

5. Security

Security in a network using MPLS should be relatively similar to security in a normal IP network.

Routing in an MPLS network uses precisely the same IP routing protocols as are currently used with IP. This implies that route filtering is unchanged from current operation. Similarly, the security of the routing protocols is not effected by the use of MPLS.

Packet filtering also may be done as in normal IP. This will require either (i) that label switching be terminated prior to any firewalls performing packet filtering (in which case a separate instance of label switching may optionally be started after the firewall); or (ii) that firewalls "look past the labels", in order to inspect the entire IP packet contents. In this latter case note that the label may imply semantics greater than that contained in the packet header: In particular, a particular label value may imply that the packet is to take a particular path after the firewall. In environments in which this is considered to be a security issue it may be desirable to terminate the label prior to the firewall.

Note that in principle labels could be used to speed up the operation of firewalls: In particular, the label could be used as an index into a table which indicates the characteristics that the packet needs to have in order to pass through the firewall. Depending upon implementation considerations matching the contents of the packet to the contents of the table may be quicker than parsing the packet in the absence of the label.

References

[ARCH]   "Multiprotocol Label Switching Architecture", E.
                Rosen, A. Viswanathan, R. Callon, work in progress,
                <draft-ietf-mpls-arch-06.txt>, August 1999.

Callon et al.   Expires March 2000   [Page 66]
□
IETF Draft   A Framework for MPLS   September 1999

[ARIS]   "ARIS: Aggregate Route-Based IP Switching", A.
                Viswanathan, N. Feldman, R. Boivie, R. Woundy, IBM
                Technical Report TR 29.2353, February 1998.

[ARIS-PROT]   "ARIS Protocol Specification", N. Feldman, A.
                Viswanathan, IBM Technical Report TR 29.2368, March
                1998.

[ATM]   "MPLS using LDP and ATM VC Switching", B. Davie, P.
                Doolan, J. Lawrence, K. McGloghrie, Y. Rekhter, E.
                Rosen, G. Swallow, work in progress, Internet Draft
                <draft-ietf-mpls-atm-02.txt>, April 1999.

[ATMVP]   "MPLS using ATM VP Switching", N. Feldman, B.
                Jamoussi, S. Komandur, A. Viswanathan, T. Worster,
                work in progress, Internet Draft <draft-feldman-mpls-
                atmvp-00.txt>, February, 1999.

[CR-LDP]   "Constraint-Based LSP Setup using LDP", B. Jamoussi,
                et. al., work in progress, <draft-ietf-mpls-cr-ldp-
                02.txt>, August 1999.

[ENCAP]   "MPLS Label Stack Encoding", E. Rosen, Y. Rekhter, D.
                Tappan, D. Farinacci, G. Fedorkow, T. Li, A. Conta,
                work in progress, Internet Draft <draft-ietf-mpls-
                label-encaps-07.txt>, September 1999.

[FANP]   "Internetworking Based on Cell Switch Router-
                Architecture and Protocol Overview", Y. Katsube, K.
                Nagami, S. Matsuzawa, H. Esaki, Proceedings of the
                IEEE, Vol. 85, No. 12, December, 1997.

[FR]   "Use of Label Switching on Frame Relay Networks", A.
                Conta, P. Doolan, A. Malis, work in progress, Internet
                Draft <draft-ietf-mpls-fr-03.txt>, November 1998.

[IPNAV]   "IP Switching for Scalable IP Services", H. Ahmed, R.
                Callon, A. Malis, J. Moy, Proceedings of the IEEE,
                Vol. 85, No. 12, December 1997.

[LDP]   "LDP Specification", L. Anderson, P. Doolan, N.
                Feldman, A. Fredette, B. Thomas, work in progress,
                <draft-ietf-mpls-ldp-05.txt>, June 1999.

[LOOP-COLOR]   "MPLS Loop Prevention Mechanism", Y. Ohba, Y.
                Katsube, E. Rosen, P. Doolan, work in progress,
                Internet Draft <draft-ietf-mpls-loop-prevention-
                01.txt>, May 1999.

Callon et al.   Expires March 2000   [Page 67]

[PNNI]   "ATM Forum Private Network-Network Interface
              Specification, Version 1.0", ATM Forum af-pnni-
              0055.000, March 1996.

[RFC1583]   "OSPF version 2", J. Moy, RFC 1583, March 1994.

[RFC1663]   "Integrated Services in the Internet Architecture: an
              Overview", R. Braden et al, RFC 1633, June 1994.

[RFC1771]   "A Border Gateway Protocol 4 (BGP-4)", Y. Rekhter, T.
              Li, RFC1771, March 1995.

[RFC1953]   "Ipsilon Flow Management Protocol Specification for
              IPv4 Version 1.0", P. Newman et al., RFC 1953, May
              1996.

[RFC2098]   "Toshiba's Router Architecture Extensions for ATM:
              Overview", Y. Katsube, K. Nagami, H. Esaki, RFC2098.

[RFC2105]   "Cisco Systems' Tag Switching Architecture Overview",
              Y. Rekhter, B. Davie, D. Katz, E. Rosen, G. Swallow,
              RFC2105, February, 1997.

[RFC2205]   "Resource ReSerVation Protocol (RSVP) Version 1
              Functional Specification", R. Braden, L. Zhang, S.
              Berson, S. Herzog, S. Jamin, RFC2205, September 1997.

[RFC2332]   "NBMA Next Hop Resolution Protocol", J. Luciani, D.
              Katz, D. Piscitello, B. Cole, N. Doraswamy, RFC2332,
              USC/Information Sciences Institute, April 1998

[RSVP-LSP]   "Extensions to RSVP for LSP Tunnels", D. Awduche, L.
              Berger, D. Gan, T. Li, G. Swallow, V. Srinivasan, work
              in progress, Internet Draft <draft-ietf-mpls-rsvp-lsp-
              tunnel-03.txt>, September 1999.

[TRAFENG]   "Requirements for Traffic Engineering Over MPLS", D.
              Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J.
              McManus, work in progress, Internet Draft <draft-ietf-
              mpls-traffic-eng-01.txt>, June 1999.

Author's Addresses

      Ross Callon
      IronBridge Networks
      55 Hayden Avenue,
      Lexington, MA 02173
      781-402-8017
      rcallon@ironbridgenetworks.com

IETF Draft   A Framework for MPLS   September 1999

      Paul Doolan
      Ennovate Networks

60 Codman Hill Road
Boxborough, MA 01719
978-263-2002
pdoolan@ennovatenetworks.com

Nancy Feldman
IBM Corp.
30 Saw Mill River Rd.
Hawthorne NY 10532
914-784-3254
nkf@us.ibm.com

Andre Fredette
Nortel Networks
3 Federal Street
Billerica, MA 01821
978-288-8524
fredette@nortelnetworks.com

George Swallow
Cisco Systems, Inc
250 Apollo Drive
Chelmsford, MA 01824
508-244-8143
swallow@cisco.com

Arun Viswanathan
Lucent Technologies
101 Crawford Corner Rd., #4D-537
Holmdel, NJ 07733
732-332-5163
arunv@dnrc.bell-labs.com

IETF Draft   A Framework for MPLS   September 1999

Network Working Group   R. Callon
Internet Draft   Ironbridge Networks
Expires: March 2000   P. Doolan
                                                Ennovate Networks
                                                     N. Feldman
                                                            IBM
                                                    A. Fredette
                                                 Nortel Networks
                                                     G. Swallow
                                                  Cisco Systems
                                                A. Viswanathan
                                            Lucent Technologies

                                                 September 1999

            A Framework for Multiprotocol Label Switching
                 <draft-ietf-mpls-framework-05.txt>

Status of this Memo

Abstract

    This document discusses technical issues and requirements for the
    Multiprotocol Label Switching working group. It is the intent of
    this document to produce a coherent description of all significant
    approaches which were and are being considered by the working

Callon et al.   Expires March 2000   [Page 1]
☐
IETF Draft   A Framework for MPLS   September 1999

    group. Selection of specific approaches, making choices regarding
    engineering tradeoffs, and detailed protocol specification, are
    outside of the scope of this framework document.

Acknowledgments

The ideas and text in this document have been collected from a
number of sources and comments received. We would like to thank
Rick Boivie, Eric Gray, Jim Luciani, Andy Malis, Rayadurgam
Ravikanth, Yakov Rekhter, Eric Rosen, Vijay Srinivasan, and Pasi
Vananen for their inputs and ideas.

1. Introduction and Requirements

1.1 Overview of MPLS

The primary goal of the MPLS working group is to standardize a
base technology that integrates the label switching forwarding
paradigm with network layer routing. This base technology (label
switching) is expected to improve the price/performance of network
layer routing, improve the scalability of the network layer, and
provide greater flexibility in the delivery of (new) routing
services (by allowing new routing services to be added without a
change to the forwarding paradigm).

The initial MPLS effort will be focused on IPv4. However, the core
technology will be extendible to multiple network layer protocols
(e.g., Ipv6, IPX, Appletalk, DECnet, CLNP). MPLS is not confined
to any specific link layer technology, it can work with any media
over which Network Layer packets can be passed between network
layer entities.

MPLS provides connection-oriented (label based) switching based on
IP routing and control protocols. MPLS may be likened to a 'shim-
layer' which is used to provide connection services to IP and
which itself makes use of link-layer services from L2 (e.g. PPP,
ATM, Ethernet).

MPLS makes use of a routing approach whereby the normal mode of
operation is that L3 routing (e.g., existing IP routing protocols
and/or new IP routing protocols) is used by all nodes to determine
the routed path. MPLS provides a simple "core" set of mechanisms
which can be applied in several ways to provide a rich
functionality. The core effort includes:

a) Semantics assigned to a stream label:

   - Labels are associated with specific streams of data.

Callon et al.   Expires March 2000   [Page 2]
□
IETF Draft   A Framework for MPLS   September 1999

b) Forwarding Methods:

   - Forwarding is simplified by the use of short fixed length
     labels to identify streams.

   - Forwarding may require simple functions such as looking
     up a label in a table, swapping labels, and possibly
     decrementing and checking a TTL.

   - In some cases, MPLS may make direct use of underlying
     layer 2switching, such as is provided by ATM [ATM] or

Frame Relay [FR] equipment.

c) Label Distribution Methods:

- Allow nodes to determine which labels to use for
  specific streams.

- This may use some sort of control exchange, and/or be
  piggybacked on a routing protocol.

The MPLS working group will define the procedures and protocols
used to assign significance to the forwarding labels and to
distribute that information between cooperating MPLS forwarders.

1.2 Requirements

- MPLS forwarding MUST simplify packet forwarding in order to
  do the following:

  o lower cost of high speed forwarding
  o improve forwarding performance

- MPLS core technologies MUST be general with respect to data
  link technologies (ie, work over a very wide range of
  underlying data links). Specific optimizations for
  particular media MAY be considered.

- MPLS core technologies MUST be compatible with a wide range
  of routing protocols, and MUST be capable of operating
  independently of the underlying routing protocols. It has
  been observed that considerable optimizations can be
  achieved in some cases by small enhancements of existing
  protocols. Such enhancements MAY be considered in the case
  of IETF standard routing protocols, and if appropriate,
  coordinated with the relevant working group(s).

- Routing protocols which are used in conjunction with MPLS
  might be based on distributed computation. As such, during

Callon et al.   Expires March 2000   [Page 3]
□
IETF Draft   A Framework for MPLS   September 1999

routing transients, these protocols may compute forwarding
paths which potentially contain loops. MPLS MUST provide
protocol mechanisms to either prevent the formation of loops
and /or contain the amount of (networking) resources that
can be consumed due to the presence of loops.

- MPLS forwarding MUST allow "aggregate forwarding" of user
  data; ie, allow streams to be forwarded as a unit and ensure
  that an identified stream takes a single path, where a
  stream may consist of the aggregate of multiple flows of
  user data. MPLS SHOULD provide multiple levels of
  aggregation support (e.g., from individual end to end
  application flows at one extreme, to aggregates of all flows
  passing through a specified switch or router at the other
  extreme).

- MPLS MUST support operations, administration, and
  maintenance facilities at least as extensive as those
  supported in current IP networks. Current network management
  and diagnostic tools SHOULD continue to work in order to

provide some backward compatibility. Where such tools are broken by MPLS, hooks MUST be supplied to allow equivalent functionality to be created.

- MPLS core technologies MUST work with both unicast and multicast streams.

- The MPLS core specifications MUST clearly state how MPLS operates in a hierarchical network.

- Scalability issues MUST be considered and analyzed during the definition of MPLS. Very scaleable solutions MUST be sought.

- MPLS core technologies MUST be capable of working with O(n) streams to switch all best-effort traffic, where n is the number of nodes in a MPLS domain. MPLS protocol standards MUST be capable of taking advantage of hardware that supports stream merging where appropriate. Note that O(n-squared) streams or VCs might also be appropriate for use in some cases.

- The core set of MPLS standards, along with existing Internet standards, MUST be a self-contained solution. For example, the proposed solution MUST NOT require specific hardware features that do not commonly exist on network equipment at the time that the standard is complete. However, the solution MAY make use of additional optional hardware features (e.g., to optimize performance).

Callon et al.   Expires March 2000   [Page 4]
□
IETF Draft   A Framework for MPLS   September 1999

- The MPLS protocol standards MUST support multipath routing and forwarding.

- MPLS MUST be compatible with the IETF Integrated Services Model, including RSVP [RFC1663][RFC2205].

- It MUST be possible for MPLS switches to coexist with non MPLS switches in the same switched network. MPLS switches SHOULD NOT impose additional configuration on non-MPLS switches.

- MPLS MUST allow "ships in the night" operation with existing layer 2 switching protocols (e.g., ATM Forum Signaling) (ie, MPLS must be capable of being used in the same network which is also simultaneously operating standard layer 2 protocols).

- The MPLS protocol MUST support both topology-driven and traffic/request-driven label assignments.

1.3 Terminology

aggregate stream

   synonym of "stream"

DLCI

a label used in Frame Relay networks to identify frame
relay circuits

flow

a single instance of an application to application flow of
data (as in the RSVP and IFMP use of the term "flow")

forwarding equivalence class

a group of L3 packets which are forwarded in the same
manner (e.g., over the same path, with the same forwarding
treatment); a forwarding equivalence class is therefore the
set of L3 packets which could safely be mapped to the same
label; note that there may be reasons that packets from a
single forwarding equivalence class may be mapped to
multiple labels (e.g., when stream merge is not used)

frame merge

stream merge, when it is applied to operation over frame
based media, so that the potential problem of cell

Callon et al.   Expires March 2000   [Page 5]

IETF Draft   A Framework for MPLS   September 1999

interleave is not an issue

label

a short fixed length physically contiguous locally
significant identifier which is used to identify a stream

label information base

the database of information containing label bindings

label stack

an ordered set of labels

label swap

the basic forwarding operation consisting of looking up an
incoming label to determine the outgoing label,
encapsulation, port, and other data handling information

label switching

a forwarding paradigm allowing streamlined forwarding of
data by using labels to identify streams of data to be
forwarded

label switched hop

the hop between two MPLS nodes, on which forwarding is done
using labels

label switched path

the path created by the concatenation of one or more label
switched hops, allowing a packet to be forwarded by

swapping labels from an MPLS node to another MPLS node

label switching router (LSR)

an MPLS node which is capable of forwarding native L3
packets

layer 2

the protocol layer under layer 3 (which therefore offers
the services used by layer 3); forwarding, when done by the
swapping of short fixed length labels, occurs at layer 2
regardless of whether the label being examined is an ATM
VPI/VCI, or a frame relay DLCI

Callon et al.   Expires March 2000   [Page 6]
▢
IETF Draft   A Framework for MPLS   September 1999

layer 3

the protocol layer at which IP and its associated routing
protocols operate

link layer

synonymous with layer 2

loop detection

a method in which loop setup may occur and data may be
injected into the loop but a mechanism is provided to
detect and break such loops

loop prevention

a method of dealing with loops in which data is never
transmitted over a loop

loop survival

a method of dealing with loops in which data may be
transmitted over a loop, but means are employed to limit
the amount of network resources which may be consumed by
the looping data

merge point

the node at which multiple streams and switched paths are
combined into a single stream sent over a single path; in
the case that the multiple paths are not combined prior to
the egress node, then the egress node becomes the merge
point

MPLS core standards

the standards which describe the core MPLS technology

MPLS domain

a contiguous set of nodes which operate MPLS routing and
forwarding and which are also in one Routing or

Administrative Domain

MPLS edge node

an MPLS node that connects an MPLS domain with a node which
is outside of the domain, either because it does not run

Callon et al.   Expires March 2000   [Page 7]
□
IETF Draft   A Framework for MPLS   September 1999

MPLS, and/or because it is in a different domain; note that
if an LSR has a neighboring host which is not running MPLS,
that LSR is an MPLS edge node

MPLS egress node

an MPLS edge node in its role in handling traffic as it
leaves an MPLS domain

MPLS ingress node

an MPLS edge node in its role in handling traffic as it
enters an MPLS domain

MPLS label

a label placed in a short MPLS shim header used to identify
streams

MPLS node

a node which is running MPLS. An MPLS node will be aware of
MPLS control protocols, will operate one or more L3 routing
protocols, and will be capable of forwarding packets based
on labels; an MPLS node may optionally be also capable of
forwarding native L3 packets (see LSR)

MultiProtocol Label Switching

an IETF working group and the effort associated with the
working group

network layer

synonymous with layer 3

shortcut VC

a VC set up as a result of an NHRP query and response

stack

synonymous with label stack

stream

an aggregate of one or more flows, treated as one flow for
the purpose of forwarding in L2 and/or L3 nodes (e.g., may
be described using a single label); in many cases a stream
may be the aggregate of a very large number of flows.

    Synonymous with "aggregate stream"

  stream merge

    the merging of several smaller streams into a larger
    stream, such that for some or all of the path the larger
    stream can be referred to using a single label

  switched path

    synonymous with label switched path

  VC merge

    stream merge when it is specifically applied to VCs,
    specifically so as to allow multiple VCs to merge into one
    single VC

  virtual circuit

    circuit used by a connection-oriented layer 2 technology
    such as ATM or Frame Relay, requiring the maintenance of
    state information in layer 2 switches

  VP merge

    stream merge when it is applied to VPs, specifically so as
    to allow multiple VPs to merge into one single VP. In this
    case the VCIs need to be unique; this allows cells from
    different sources to be distinguished via the VCI

  VPI/VCI

    a label used in ATM networks to identify ATM virtual
    circuits

1.4 Acronyms and Abbreviations

    DLCI   Data Link Circuit Identifier

    FEC   Forwarding Equivalence Class

    ISP   Internet Service Provider

    LIB   Label Information Base

    LDP   Label Distribution Protocol

    L2   Layer 2

    L3   Layer 3

LSP	Label Switched Path
LSR	Label Switching Router
MPLS	MultiProtocol Label Switching
MPT	Multipoint to Point Tree
NHC	Next Hop (NHRP) Client
NHS	Next Hop (NHRP) Server
VC	Virtual Circuit
VCI	Virtual Circuit Identifier
VPI	Virtual Path Identifier

1.5 Motivation for MPLS

This section describes the expected and potential benefits of the MPLS over existing schemes. Specifically, this section discusses the advantages of MPLS over previous methods for building core networks (ie, networks for internet service providers or for major corporate backbones). The potential advantages of MPLS in campus and local area networks are not discussed in this section.

There are currently two commonly used methods for building core IP networks: (i) Networks of datagram routers in which the core of the network is based on the datagram routers; (ii) Networks of datagram routers operating over an ATM core. In order to describe the advantages of MPLS, it is necessary to know which alternate to MPLS we are using for the comparison. This section is therefore split into two sections: Section 1.5.1 describes the advantages of MPLS when compared to a pure datagram routed network. Section 1.5.2 describes the advantages of MPLS when compared to an IP over ATM network.

This section does not provide a complete list of requirements for MPLS. For example, Multipoint to Point Trees are important for MPLS to scale. However, datagram forwarding naturally acts in this way (since multiple sources are merged automatically), and the ATM forum is currently adding support for multipoint to point to the ATM standards. The ability to do MPTs is therefore important to MPLS, but does not represent an advantage over either datagram routing or IP over ATM, and therefore is not mentioned in this section.

Callon et al.   Expires March 2000   [Page 10]

IETF Draft   A Framework for MPLS   September 1999

1.5.1 Benefits Relative to Use of a Router Core

1.5.1.1 Simplified Forwarding

Label switching allows packet forwarding to be based on an exact match for a short label, rather than a longest match algorithm applied to a longer address as is required for normal datagram forwarding. In addition, the label headers used with MPLS are simpler than the headers typically used with datagram protocols

such as IP. This in turn implies that MPLS allows a much simpler forwarding paradigm relative to datagrams, and implies that it is easier to build a high speed router using MPLS.

Whether this simpler forwarding operation will result in availability of LSRs which can operate at higher speeds than datagram routers is controversial, and probably depends upon implementation details. There are some parts of the network, such as at hierarchical boundaries, where datagram IP forwarding at high speed will be required. This implies that implementation of a high speed router is highly desirable. In addition, there are currently multiple companies building high speed routers which will allow IP packets to be forwarded at very high speed. At speeds at least up to OC48, it appears that once the one-time engineering is completed, the per-unit cost associated with IP forwarding will be a small fraction of the overall equipment cost.

However, there are also many existing routers which can benefit from the simpler forwarding allowed by MPLS. In addition, there are some routers being built with implementations that will benefit from the simpler forwarding available with MPLS.

1.5.1.2 Efficient Explicit Routing

Explicit routing (aka Source Routing) is a very powerful technique which potentially can be useful for a variety of purposes. However, with pure datagram routing the overhead of carrying a complete explicit route with each packet is prohibitive. However, MPLS allows the explicit route to be carried only at the time that the label switched path is set up, and not with each packet. This implies that MPLS makes explicit routing practical. This in turn implies that MPLS can make possible a number of advanced routing features which depend upon explicit routing.

1.5.1.3 Traffic Engineering

Traffic engineering refers to the process of selecting the paths chosen by data traffic in order to balance the traffic load on the various links, routers, and switches in the network. Traffic engineering is most important in networks where multiple parallel or alternate paths are available. The rapid growth in the

Callon et al.   Expires March 2000   [Page 11]
IETF Draft   A Framework for MPLS   September 1999

Internet, and particularly the associated rapid growth in the demand for bandwidth, has tended to cause some core networks to become increasingly "branchy" in recent years, resulting in an increase in the importance of traffic engineering [TRAFENG].

It is common today, in networks that are running IP over an ATM core using PVCs, to manually configure the path of each PVC in order to equalize the traffic levels on different links in the network. Thus traffic engineering is typically done today in IP over ATM networks using manual configuration.

Traffic engineering is difficult to accomplish with datagram routing. Some degree of load balancing can be obtained by adjusting the metrics associated with network links. However, there is a limit to how much can be accomplished in this way, and in networks with a large number of alternative paths between any two points balancing of the traffic levels on all links is

difficult to achieve solely by adjustment of the metrics used with hop by hop datagram routing.

MPLS allows streams from any particular ingress node to any particular egress node to be individually identified. MPLS therefore provides a straightforward mechanism to measure the traffic associated with each ingress node to egress node pair. In addition, since MPLS allows efficient explicit routing of Label Switched Paths, it is straightforward to ensure that any particular stream of data takes the preferred path.

The hard part of traffic engineering is selection of the method used to route each Label Switched Path. There are a variety of possible ways to do this, ranging from manual configuration of routes, to use of a routing protocol which announces traffic loads in the network combined with background recomputation of paths.

1.5.1.4 QoS Routing

QoS routing refers to a method of routing in which the route chosen for a particular stream is chosen in response to the QoS required for that stream. In many cases QoS routing needs to make use of explicit routing for several reasons:

In some cases specific bandwidth is likely to be reserved for each of many specific streams of data. This implies that the total bandwidth of multiple streams may exceed the bandwidth available on any particular link, and thus not all streams, even between the same ingress and egress nodes, can take the same path. Instead, individual streams will need to be individually routed. This is somewhat analogous to traffic engineering, but might require separation of streams on a finer granularity. Thus explicit routing may be needed in order to allow each stream to be

Callon et al.   Expires March 2000   [Page 12]
□
IETF Draft   A Framework for MPLS   September 1999

individually routed, and to eliminate the need for each switch along the path of a stream to compute the route for each stream.

Consider the case of routing a stream with a specific bandwidth requirement: In this case the route chosen will depend upon the amount of bandwidth which is requested. For any one given bandwidth, it is straightforward to select a path. However there are a lot of different levels of bandwidth which could in principle be requested. This makes it impractical to precompute all possible paths for all possible bandwidths. If the path for a particular stream must be computed on demand, then it is undesirable to require every LSR on the path to compute the path. Instead, it is preferable to have the first node compute the path and specify the route to be followed through use of an explicit route.

For a variety of reasons the information available for QoS routing may in some cases be slightly out of date. This implies that the attempt to select a specific path for a QoS-sensitive stream may in some cases fail, due to a particular node or link not having the required resources available. In these cases it is not in general always feasible to tell all other nodes in the network of the limited resource in one particular network element. If explicit routing is available, then this permits the initial node of the stream (the ingress node in MPLS) to be informed that the

indicated network element is not able to carry the stream,
allowing an alternate path to be selected. However, in this case
the node that selects the alternate path has to use explicit
routing in order to force the stream to follow the alternate path.

These and similar examples imply that explicit routing is
necessary in order to do an adequate job of QoS routing. Given
that MPLS allows efficient explicit routing, it follows that MPLS
also facilitates QoS routing.

1.5.1.5 Mappings from IP Packet to Forwarding Equivalence Class

MPLS allows the mapping from IP packet to forwarding equivalence
class to be performed only once, at the ingress to an MPLS domain.
This facilitates complex mappings from IP packet to FEC that would
otherwise be impractical.

For example, consider the case of provisioned QoS: Some ISPs offer
a service wherein specific customers subscribe to receive
differentiated services (e.g., their packets may receive
preferential forwarding treatment). Mapping of IP packets to the
service level may require knowing the customer who is transmitting
the packet, which may in turn require packet filtering based on
source and destination address, incoming interface, and other
characteristics. The sheer number of filters that are needed in a

口
IETF Draft   A Framework for MPLS   September 1999

moderate sized ISP preclude repetition of the filters at every
router throughout the network. Also, some information such as
incoming interface is not available except at the ingress node to
the network. This implies that the preferred way to offer
provisioned QoS is to map the packet at the ingress point to the
preferred QoS level, and then label the packet in some way. MPLS
offers an efficient method to label the QoS class associated with
any particular packet.

Other examples of complex mappings from IP packet to FEC are also
likely to be determined as MPLS is deployed.

1.5.1.6 Partitioning of Functionality

Due to the support of the different label granularities, it will
be possible to hierarchically partition the processing
functionality to the different network elements, so that the more
heavy processing takes place on the edges of the network, near the
customers, and on the core network the processing is as simple as
possible, e.g. pure label based forwarding.

AS level aggregations will enable building of the fully switched
backbone networks and traffic exchange points. Also, it will be
possible for operators to fully switch the transit traffic
traveling through the operator's network. Deaggregation will be
needed for the streams that are destined in the networks connected
to the MPLS domain, but it shall be noted that this deaggregation
will only need to perform lookup operations associated with
finding the label for egress router or interface, e.g. TOS
information bound to label in source is still valid, and can be
honored on basis of which label the packet was received in. It
shall be noted that it is even impossible for the receiving domain
to do the classification as the original packet classification

policy is not known by the receiving domain.

As one example of the improved functional partitioning, consider
the case of the use of packet filters to map IP packets into a
substantial number of queues, such that each queue receives
differentiated services. For example, suppose that a network
supports individual queuing for on the order of 100 different
customers, with packets mapped to queues based on the source and
destination IP address. In this case, with MPLS the packet
filtering can be done solely on the edge of the network, with the
packets mapped to labels such that each individual user receives
separate labels. Thus with MPLS the filtering can be performed at
the edge only of the network. This allows complex mappings of IP
packets to forwarding equivalence class.

1.5.1.7 Single Forwarding Paradigm with Service Level Differentiation

Callon et al.   Expires March 2000   [Page 14]
□
IETF Draft   A Framework for MPLS   September 1999

MPLS can allow a single forwarding paradigm to be used to support
multiple types of service on the same network.

Because of the forwarding paradigm, it will be possible to carry
the different services through the same network elements,
regardless of the control plane protocols used for the population
of the LSR's LIB. It is for example possible, in case of ATM based
switching system to support all the native ATM services, frame
relay services, and labeled IP services. The simultaneous support
of multiple service may need partitioning of the label space
between the services, and shall be supported by the label
distribution management protocol.

Non-exhaustive list of examples of the services suitable for
carrying over LSRs are IP traffic, Frame Relay traffic, ATM
traffic (in case of cell switching), IP tunneling, VPNs, and other
datagram protocols.

Note that MPLS does not necessarily use the same header format
over all types of media. However, over any particular type of
media a single header format (at least for the lowest level of the
Label Stack) should be possible.

1.5.2 Benefits Relative to Use of an ATM or Frame Relay Core

Note: This section compares MPLS with other methods for
interconnecting routers over a switched core network. We are not
considering methods for interconnecting hosts located on virtual
networks. For example the ATM Forum LANE and MPOA standards
support virtual networks. MPLS does not directly support virtual
networks, and should not be compared directly with MPOA or LANE.

Previously available methods for interconnecting routers in an IP
over ATM environment make use of either: (i) a full mesh 'n-
squared' overlay of virtual circuits between n ATM-attached
routers; (ii) A partial mesh of VCs between routers; or (iii) A
partial mesh of VCs, plus the use of NHRP to facilitate on demand
cut-through SVCs.

1.5.2.1 Scaling of the Routing Protocol

Relative to the interconnection of IP over an ATM core, MPLS
improves the scaling of routing due to reduced number of peers and
elimination of the 'n-squared' logical links between routers used
to operate the routing protocols.

Because all LSRs will run standard routing protocols, the number
of the peers routers need to communicate with is reduced to the
number of LSRs and routers a given LSR is directly connected to,
instead of having to peer with large numbers of routers at the

Callon et al.   Expires March 2000   [Page 15]
☐
IETF Draft   A Framework for MPLS   September 1999

ends of the switched L2 paths. This benefit is achieved because
the edge LSRs do not need to peer with every other edge LSR in the
domain as is the case on a hybrid switch / router network."

1.5.2.2 Common Operation over Packet and Cell media

MPLS makes use of common methods for routing and forwarding over
packet and cell media, and potentially allows a common approach to
traffic engineering, QoS routing, and other aspects of operation.
For example, this means that the same method for label
distribution can be used over Frame Relay and ATM media, as well
as between LSRs using the MPLS Shim Header for forwarding over
other media (such as PPP links and broadcast LANs).

Note: There may be some differences with respect to the operation
of different media. For example, if VP merge is used with ATM
media (rather than VC merge) then the merge operation may be
somewhat different than what it would be with packet media or with
ATM using VC merge.

1.5.2.3 Easier Management

The use of a common method for label distribution and common
routing protocols over multiple types of media is expected to
simplify network management of MPLS networks.

1.5.2.4 Elimination of the 'Routing over Large Clouds' Issue

MPLS eliminates the need to use NHRP and on-demand cut-through
SVCs for operation over ATM. This eliminates the latency problem
associated with cut-through SVCs.

2. Discussion of Core MPLS Components

2.1 The Basic Routing Approach

Routing is accomplished through the use of standard L3 routing
protocols, such as OSPF and BGP [RFC1583][RFC1771]. The
information maintained by the L3 routing protocols is then used to
distribute labels to neighboring nodes that are used in the
forwarding of packets as described below. In the case of ATM
networks, the labels that are distributed are VPI/VCIs and a
separate protocol (ie, PNNI) is not necessary for the
establishment of VCs for IP forwarding.

The topological scope of a routing protocol (ie routing domain)
and the scope of label switching MPLS-capable nodes may be
different. For example, MPLS-knowledgeable and MPLS-ignorant

nodes, all of which are OSPF routers, may be co-resident in an

area. In the case that neighboring routers know MPLS, labels can
be exchanged and used.

Neighboring MPLS routers may use configured PVCs or PVPs to tunnel
through non-participating ATM or FR switches.

2.2 Labels

In addition to the single routing protocol approach discussed
above, the other key concept in the basic MPLS approach is the use
of short fixed length labels to simplify user data forwarding.

2.2.1 Label Semantics

It is important that the MPLS solutions are clear about what
semantics (ie, what knowledge of the state of the network) is
implicit in the use of labels for forwarding user data packets or
cells.

At the simplest level, a label may be thought of as nothing more
than a shorthand for the packet header, in order to index the
forwarding decision that a router would make for the packet. In
this context, the label is nothing more than a shorthand for an
aggregate stream of user data.

This observation leads to one possible very simple interpretation
that the "meaning" of the label is a strictly local issue between
two neighboring nodes. With this interpretation: (i) MPLS could be
employed between any two neighboring nodes for forwarding of data
between those nodes, even if no other nodes in the network
participate in MPLS; (ii) When MPLS is used between more than two
nodes, then the operation between any two neighboring nodes could
be interpreted as independent of the operation between any other
pair of nodes. This approach has the advantage of semantic
simplicity, and of being the closest to pure datagram forwarding.
However this approach (like pure datagram forwarding) has the
disadvantage that when a packet is forwarded it is not known
whether the packet is being forwarded into a loop, into a black
hole, or towards links which have inadequate resources to handle
the traffic flow. These disadvantages are necessary with pure
datagram forwarding, but are optional design choices to be made
when label switching is being used.

There are cases where it would be desirable to have additional
knowledge implicit in the existence of the label. For example, one
approach to avoiding loops (see section 4.3) involves signaling
the label distribution along a path before packets are forwarded
on that path. With this approach the fact that a node has a label
to use for a particular IP packet would imply the knowledge that
following the label (including label switching at subsequent

nodes) leads to a non-looping path which makes progress towards
the destination (something which is usually, but not necessarily
always true when using pure datagram routing). This would of
course require some sort of label distribution/setup protocol
which signals along the path being setup before the labels are
available for packet forwarding. However, there are also other
consequences to having additional semantics associated with the
label: specifically, procedures are needed to ensure that the
semantics are correct. For example, if the fact that you have a
label for a particular destination implies that there is a loop-
free path, then when the path changes some procedures are required
to ensure that it is still loop free. Another example of semantics
which could be implicit in a label is the identity of the higher
level protocol type which is encoded using that label value.

In either case, the specific value of a label to use for a stream
is strictly a local issue; however the decision about whether to
use the label may be based on some global (or at least wider
scope) knowledge that, for example, the label-switched path is
loop-free and/or has the appropriate resources.

A similar example occurs in ATM networks: With standard ATM a
signaling protocol is used which both reserves resources in
switches along the path, and which ensures that the path is loop-
free and terminates at the correct node. Thus implicit in the fact
that an ATM node has a VPI/VCI for forwarding a particular piece
of data is the knowledge that the path has been set up
successfully.

Another similar example occurs with multipoint to point trees over
ATM (see section 4.2 below), where the multipoint to point tree
uses a VP, and cell interleave at merge points in the tree is
handled by giving each source on the tree a distinct VCI within
the VP. In this case, the fact that each source has a known
VPI/VCI to use needs to (implicitly or explicitly) imply the
knowledge that the VCI assigned to that source is unique within
the context of the VP.

In general labels are used to optimize how the system works, not
to control how the system works. For example, the routing protocol
determines the path that a packet follows. The presence or absence
of a label assignment should not affect the path of a L3 packet.
Note however that the use of labels may make capabilities such as
explicit routes, loadsharing, and multipath more efficient.

2.2.2 Label Granularity

Labels are used to create a simple forwarding paradigm. The
essential element in assigning a label is that the device which
will be using the label to forward packets will be forwarding all

packets with the same label in the same way. If the packet is to
be forwarded solely by looking at the label, then at a minimum,
all packets with the same incoming label should be forwarded out
the same port(s) with the same encapsulation(s), and with the same
next hop label if any (although the special cases of multipath and
load sharing are an exception to this rule).

The term "forwarding equivalence class" is used to refer to a set
of L3 packets which are all forwarded in the same manner by a
particular LSR (for example, the IP packets in a forwarding
equivalence class may be destined for the same egress from an MPLS
network, and may be associated with the same QoS class). A
forwarding equivalence class is therefore the set of L3 packets
which could safely be mapped to the same label. Note that there
may be reasons that packets from a single forwarding equivalence
class may be mapped to multiple labels (e.g., when stream merge is
not used).

Note that the label could also mean "ignore this label and forward
based on what is contained within," where within one might find a
label (if a stack of labels is used) or a layer 3 packet.

For IP unicast traffic, the granularity of a label allows various
levels of aggregation in a Label Information Base (LIB). At one
end of the spectrum, a label could represent a host route (ie the
full 32 bits of IP address). If a router forwards an entire CIDR
prefix in the same way, it may choose to use one label to
represent that prefix. Similarly if the router is forwarding
several (otherwise unrelated) CIDR prefixes in the same way it may
choose to use the same label for this set of prefixes. For
instance all CIDR prefixes which share the same BGP Next Hop could
be assigned the same label. Taking this to the limit, an egress
router may choose to advertise all of its prefixes with the same
label.

By introducing the concept of an egress identifier, the
distribution of labels associated with groups of CIDR prefixes can
be simplified. For instance, an egress identifier might specify
the BGP Next Hop, with all prefixes routed to that next hop
receiving the label associated with that egress identifier.
Another natural place to aggregate would be the MPLS egress
router. This would work particularly well in conjunction with a
link-state routing protocol, where the association between egress
router and CIDR prefix is already distributed throughout an area.

For IP multicast, the natural binding of a label would be to a
multicast tree, or rather to the branch of a tree which extends
from a particular port. Thus for a shared tree, the label
corresponds to the multicast group, (*,G). For (S,G) state, the
label would correspond to the source address and the multicast

Callon et al.   Expires March 2000   [Page 19]
□
IETF Draft   A Framework for MPLS   September 1999

group.

A label can also have a granularity finer than a host route. That
is, it could be associated with some combination of source and
destination address or other information within the packet. This
might for example be done on an administrative basis to aid in
effecting policy. A label could also correspond to all packets
which match a particular Integrated Services filter specification.

Labels can also represent explicit routes. This use is
semantically equivalent to using an IP tunnel with a complete
explicit route. This is discussed in more detail in section 4.10.

2.2.2.1 Examples of Unicast traffic granularities:

- PQ (Port Quadruples) same IP source address prefix,
  destination address prefix, TTL, IP protocol and TCP/UDP
  source/destination ports

- PQT (Port Quadruples with TOS) same IP source address
  prefix, destination address prefix, TTL, IP protocol and
  TCP/UDP source/destination ports and same IP header TOS
  field (including Precedence and TOS bits).

- HP (Host Pairs) Same specific IP source and destination
  address (32 bit)

- NP (Network Pairs) Same IP source and destination address
  prefixes (variable length)

- DN (Destination Network) Same IP destination network
  address prefix (variable length)

- ER (Egress Router) Same egress router ID (e.g. OSPF)

- NAS (Next-hop AS) Same next-hop AS number (BGP)

- DAS (Destination AS) Same destination AS number (BGP)

2.2.2.2 Multicast traffic granularities:

- SST (Source Specific Tree) Same source address and
  multicast group

- SMT (Shared Multicast Tree) Same multicast group address

2.2.3 Label Assignment

Essential to label switching is the notion of binding between a
label and Network Layer routing (routes). A control component is

Callon et al.   Expires March 2000   [Page 20]
□
IETF Draft   A Framework for MPLS   September 1999

responsible for creating label bindings, and then distributing the
label binding information among label switches. Label assignment
involves allocating a label, and then binding a label to a route.

Label assignment can be driven by control traffic or by data
traffic. This is discussed in more detail in section 3.4.

Control traffic driven label assignment has several advantages, as
compared to data traffic driven label assignment. For one thing,
it minimizes the amount of additional control traffic needed to
distribute label binding information, as label binding information
is distributed only in response to control traffic, independent of
data traffic. It also makes the overall scheme independent of and
insensitive to the data traffic profile/pattern. Control traffic
driven creation of label binding improves forwarding latency, as
labels are assigned before data traffic arrives, rather than being
assigned as data traffic arrives. It also simplifies the overall
system behavior, as the control plane is controlled solely by
control traffic, rather than by a mix of control and data traffic.

There are however situations where data traffic driven label
assignment is necessary. A particular case may occur with ATM

without VP or VC merge. In this case in order to set up a full
mesh of VCs would require n-squared VCs. However, in very large
networks this may be infeasible. Instead VCs may be setup where
required for forwarding data traffic. In this case it is generally
not possible to know a priori how many such streams may occur.

Label withdrawal is required with both control-driven and data-
driven label assignment. Label withdrawal is primarily a matter of
garbage collection, that is collecting up unused labels so that
they may be reassigned. Generally speaking, a label should be
withdrawn when the conditions that allowed it to be assigned are
no longer true. For example, if a label is imbued with extra
semantics such as loop-free-ness, then the label must be withdrawn
when those extra semantics cease to hold.

In certain cases, notably multicast, it may be necessary to share
a label space between multiple entities. If these sharing
arrangements are altered by the coming and going of neighbors,
then labels which are no longer controlled by an entity must be
withdrawn and a new label assigned.

2.2.4 Label Stack and Forwarding Operations

The basic forwarding operation consists of looking up the incoming
label to determine the outgoing label, encapsulation, port, and
any additional information which may pertain to the stream such as
a particular queue or other QoS related treatment. We refer to
this operation as a label swap.

Callon et al.   Expires March 2000   [Page 21]
口
IETF Draft   A Framework for MPLS   September 1999

When a packet first enters an MPLS domain, the packet is forwarded
by normal layer 3 forwarding operations with the exception that
the outgoing encapsulation will now include a label. We refer to
this operation as a label push. When a packet leaves an MPLS
domain, the label is removed. We refer to this as a label pop.

In some situations, carrying a stack of labels is useful. For
instance both IGP and BGP label could be used to allow routers in
the interior of an AS to be free of BGP information. In this
scenario, the "IGP" label is used to steer the packet through the
AS and the "BGP" label is used to switch between ASes.

With a label stack, the set of label operations remains the same,
except that at some points one might push or pop multiple labels,
or pop & swap, or swap & push.

2.3 Encapsulation

Label-based forwarding makes use of various pieces of information,
including a label or stack of labels, and possibly additional
information such as a TTL field [ENCAP]. In some cases this
information may be encoded using an MPLS header, in other cases
this information may be encoded in L2 headers. Note that there may
be multiple types of MPLS headers. For example, the header used
over one media type may be different than is used over a different
media type. Similarly, in some cases the information that MPLS
makes use of may be encoded in an ATM header. We will use the term
"MPLS encapsulation" to refer to whatever form is used to
encapsulate the label information and other information used for

label based forwarding. The term "MPLS header" will be used where
this information is carried in some sort of MPLS-specific header
(ie, when the MPLS information cannot all be carried in a L2
header). Whether there is one or multiple forms of possible MPLS
headers is also outside of the scope of this document.

The exact contents of the MPLS encapsulation is outside of the
scope of this document. Some fields, such as the label, are
obviously needed. Some others might or might not be standardized,
based on further study. An encapsulation scheme may make use of
the following fields:
- label
- TTL
- class of service
- stack indicator
- next header type indicator
- checksum

It is desirable to have a very short encapsulation header. For
example, a four byte encapsulation header adds to the convenience

Callon et al.   Expires March 2000   [Page 22]
□
IETF Draft   A Framework for MPLS   September 1999

of building a hardware implementation that forwards based on the
encapsulation header. But at the same time it is tricky assigning
such a limited number of bits to carry the above listed
information in an MPLS header. Hence careful consideration must be
given to the information chosen for an MPLS header.

A TTL value in the MPLS header may be useful in the same manner as
it is in IP. Specifically, TTL may be used to terminate packets
caught in a routing loop, and for other related uses such as
traceroute. The TTL mechanism is a simple and proven method of
handling such events. Another use of TTL is to expire packets in a
network by limiting their "time to live" and eliminating stale
packets that may cause problems for some of the higher layer
protocols. When used over link layers which do not provide a TTL
field, alternate mechanisms will be needed to replace the uses of
the TTL field.

A provision for a class of service (COS) field in the MPLS header
allows multiple service classes within the same label. However,
when more sophisticated QoS is associated with a label, the COS
may not have any significance. Alternatively, the COS (like QoS)
can be left out of the header, and instead propagated with the
label assignment, but this entails that a separate label be
assigned to each required class of service. Nevertheless, the COS
mechanism provides a simple method of segregating flows within a
label.

As previously mentioned, the encapsulation header can be used to
derive benefits of tunneling (or stacking).

The MPLS header must provide a way to indicate that multiple MPLS
headers are stacked (ie, the "stack indicator"). For this purpose
a single bit in the MPLS header will suffice. In addition, there
are also some benefits to indicating the type of the protocol
header following the MPLS header (ie, the "next header type
indicator"). One option would be to combine the stack indicator
and next header type indicator into a single value (ie, the next
header type indicator could be allowed to take the value "MPLS

header"). Another option is to have the next header type indicator be implicit in the label value (such that this information would be propagated along with the label).

There is no compelling reason to support a checksum field in the MPLS header. A CRC mechanism at the L2 layer should be sufficient to ensure the integrity of the MPLS header.

3. Observations, Issues and Assumptions

3.1 Layer 2 versus Layer 3 Forwarding

Callon et al.   Expires March 2000   [Page 23]
□
IETF Draft   A Framework for MPLS   September 1999

MPLS uses L2 switching as a way to provide simple and fast packet forwarding capability. One primary reason for the simplicity of L2 layer switching comes from its short, fixed length labels. A node forwarding at L3 must parse a (relatively) large header, and perform a longest-prefix match to determine a forwarding path. However, when a node performs MPLS label switching, and labels are assigned properly, it can do a direct index lookup into its forwarding (or in this case, label-switching) table with the short header. It is arguably simpler to build label switching hardware than it is to build L3 forwarding hardware because the label switching function is less complex.

The relative performance of MPLS switching and L3 forwarding may differ considerably between nodes. Some nodes may illustrate an order of magnitude difference. Other nodes (for example, nodes with more extensive L3 forwarding hardware) may have identical performance. However, some nodes may not be capable of doing a L3 forwarding at all (e.g. some ATM implementations), or have such limited capacity as to be unusable at L3. In this situation, traffic may be blackholed if no switched path exists. Note that delaying route advertisements until a switched path exists for associated packets may reduce or eliminate the need to black hole these packets.

On nodes in which L3 forwarding is slower than L2 switching, pushing traffic to L3 when no L2 path is available may cause congestion. In some cases this could cause data loss (since L3 may be unable to keep up with the increased traffic). However, if data is discarded, then in general this will cause TCP to backoff, which would allow control traffic, traceroute and other network management tools to continue to work.

The MPLS protocol MUST not make assumptions about the forwarding capabilities of an MPLS node. Thus, MPLS must propose solutions that can leverage the benefits of a node that is capable of L3 forwarding, but must not mandate the node be capable of such.

Why We Will Still Need L3 Forwarding:

MPLS will not, and is not intended to, replace L3 forwarding. There is absolutely a need for some systems to continue to forward IP packets using normal Layer 3 IP forwarding. L3 forwarding will be needed for a variety of reasons, including:
  - For scaling; to forward on a finer granularity than the
    labels can provide